NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

STATISTICAL ANALYSIS OF THE SKAION NETWORK SECURITY DATASET

by

William F. Major, Jr.

September 2012

Thesis Advisor:	Lyn R. Whitaker
Thesis Co-Advisor:	Harrison C. Schramm
Second Reader:	Richard E. Harang

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2012	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Statistical Analysis of the Skaion Network Security Dataset		**5. FUNDING NUMBERS**	
6. AUTHOR(S) William F. Major, Jr.			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number _____N/A_____.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited		**12b. DISTRIBUTION CODE** A	

13. ABSTRACT (maximum 200 words)

This thesis considers the best use of network traffic data to increase cyber security. This operational problem is one of great concern to network administrators and users generally. Our specific task was performed for the Network Security Division of the Army Research Laboratory, who requested we analyze a dataset of cyber-attacks masked in a background of representative network traffic (the "Skaion" dataset). We find that substantial preprocessing must done before statistical methods can be applied to raw network data, that no single predictor is sufficient, and that the most effective statistical analysis is logistic regression. Our approach is novel in that we consider not only single predictor events, but also combinations of reports from multiple tools. While we consider a number of different statistical techniques, we find that the most satisfactory model is based on logistic regression. Finally, we conclude that while the Skaion dataset is valuable in terms of its new approach to network traffic emulation, the 1999 KDD Cup and DARPA-MIT datasets—despite their many shortcomings—are more clearly organized and accessible to academic study. Cyber security is a globally important problem and datasets like Skaion's must maximize opportunities for cross-disciplinary academic endeavors.

14. SUBJECT TERMS Data analysis, network security, cyber security, ordinal logistic regression.			**15. NUMBER OF PAGES** 77
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

STATISTICAL ANALYSIS OF THE SKAION NETWORK SECURITY DATASET

William F. Major, Jr.
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 2001

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 2012

Author: William F. Major, Jr.

Approved by: Lyn R. Whitaker
Thesis Advisor

Harrison C. Schramm
Thesis Co-Advisor

Richard E. Harang
Second Reader

Robert F. Dell
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis considers the best use of network traffic data to increase cyber security. This operational problem is one of great concern to network administrators and users generally. Our specific task was performed for the Network Security Division of the Army Research Laboratory, who requested we analyze a dataset of cyber-attacks masked in a background of representative network traffic (the "Skaion" dataset). We find that substantial preprocessing must done before statistical methods can be applied to raw network data, that no single predictor is sufficient, and that the most effective statistical analysis is logistic regression. Our approach is novel in that we consider not only single predictor events, but also combinations of reports from multiple tools. While we consider a number of different statistical techniques, we find that the most satisfactory model is based on logistic regression. Finally, we conclude that while the Skaion dataset is valuable in terms of its new approach to network traffic emulation, the 1999 KDD Cup and DARPA-MIT datasets—despite their many shortcomings—are more clearly organized and accessible to academic study. Cyber security is a globally important problem and datasets like Skaion's must maximize opportunities for cross-disciplinary academic endeavors.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AD	Anomaly Detection
AICc	Akaike Information Criterion
AFRL	Air Force Research Laboratory
ARDA	Advanced Research and Development Agency
ARL	Army Research Laboratory
CGI	Common Gateway Interface
CSV	Comma Separated Value
CYBERCOM	Cyber Command
DARPA	Defense Advanced Research & Project Agency
DDoS	Distributed Denial of Service
DNI-U	Director of Naval Intelligence - Unclassified
DoD	Department of Defense
DOS	Denial of Service
FTP	File Transfer Protocol
GB	Gigabyte
GUI	Graphic User Interface
HTTP	Hypertext Transfer Protocol
IC	Intelligence Community
IDS	Intrusion Detection System
IP	Internet Protocol
KB	Kilobyte
KDD	Knowledge Discovery and Data Mining
MB	Megabyte

MORS	Military Operations Research Society
NIPRNET	Non-Secure Internet Protocol Network
OPNAV	Office of the Chief of Naval Operations
OR	Operations Research
OSIS	Open Source Information System
PCAP	Packet Capture
RAM	Random Access Memory
R2L	Remote to Local
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
SSH	Secure Shell
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
U2R	User to Root

EXECUTIVE SUMMARY

This thesis applies techniques of Military Operations Research to consider the optimal use of network traffic data to increase cyber security. Our specific task was performed for the Network Security Division of the Army Research Laboratory, who requested we analyze a representative dataset of cyber-attacks masked in a background of representative network traffic (the "Skaion" dataset).

The Skaion dataset was created as a test bed for staging attack simulations against notional Intelligence Community (IC) networks, which could then enable opportunities for further study and innovation for protection of valuable IC networks. Our approach is to consider which of the many predictor events in this dataset best determine when malicious activity is taking place. Our approach is novel in that we consider not only single predictor events, but also combinations of reports from multiple tools. While we consider a number of different statistical techniques, we find that the most satisfactory model is based on logistic regression.

For comparison, we researched the DARPA-MIT 1999 dataset and the 1999 KDD Cup dataset, and we find that the Skaion dataset differs from these well-known network security datasets with regard to the following four key features: 1) the two Skaion scenarios we studied are far smaller in terms of the time and amount of data captured; 2) the Skaion dataset is far less dense with attack activity; 3) the manner in which the Skaion dataset "labels" malicious activity from benign activity is unique in that it is organized by Internet Protocol (IP) address role (i.e., IPs are either attackers or normal), while the other two datasets label each conversation between two terminal IP addresses (i.e., a connection is either an attack or normal), which introduces the unrealistic implications the other data sets do not: that every conversation originating from an attacker IP is an attack, and any IP address that initiates with normal activity will never become and attacker and vice-versa; finally, 4) the Skaion data set includes predictions from intrusion detection system (IDS) alert logs and predictions of the roles of IP addresses, which offers an opportunity to explore methods of data fusion and suppression of "noise" in order to minimize and prioritize network alert analysis.

Our approach is to combine these predictions with network diagnostic alerts and basic network behavior statistics in an ordinal logistic regression to test its predictive ability relative to that of IDS alerts alone. In doing so, we find that the detection percentage increased from 29% (28 of 98 attacks detected) with the IDS alerts alone to 82.7% (74 of 98 attacks detected) with the ordinal logistic regression predictions. The number of false alarms rises from zero with the IDS alerts to only 15 in the regression model, which equates to an overall misclassification rate of 0.13% (24 missed attacks and 15 false alarms out of 10,894 connections in the validation set).

The conclusions from our work are threefold:

1) No single detection tool or statistical method is adequate to keep the network secure,

2) Given good predictors, simple models can be very powerful,

3) That network emulators and experimenters need to consider data analysis when they design their projects so that detailed statistical analysis may be done rapidly. This is the greatest opportunity for Operations Research and Network Security Analysts to bridge the gap between professions.

The operational impact of our work is immediate. By understanding the types of data that are most useful for detecting threats, the processing may be allocated optimally, and less metadata (currently 10% of network capacity) will be required. Finally, we see this work as the first of many projects that will apply OR techniques in the Cyber domain.

ACKNOWLEDGMENTS

Special thanks to my wife, Molly, for—while securing her own degree—making sure everything else was taken care of so I could finish this thesis and the rest of my studies. Your encouragement and support was essential to my success. I love you very much and I'm excited about what lies ahead for us.

To my son, Billy, and my daughter, Abby, thank you both for making the time we had together in Monterey special, even if there was far too little of it. I hope someday you will both have the chance to read this thesis and be proud of the work I did while I was away from the two of you at school. More importantly, I hope both of you go on to do much greater work yourselves.

To my military faculty advisor and Naval Aviation/Operations Research mentor, CDR Harrison Schramm, USN, I am eternally indebted to you for your generosity of time and professional expertise. Your lightning-fast turnaround kept me on task and your big picture analysis kept me on track. If my career as a naval officer and an analyst results in merely a statistically significant proportion of what you have accomplished, I will be quite satisfied.

To my civilian faculty advisor, Professor Lyn Whitaker, Ph.D., thank you for your enthusiasm, encouragement and insight. Your questions and comments were always laser-accurate and kept me from heading down fruitless rabbit holes and ignoring crucial details.

Thanks also to Professor Samuel Buttrey, Ph.D., who was kind enough to drop everything and advise on solutions for dealing with "Big Data"—especially when I missed what should have been intuitively obvious to the most casual observer (like "CTRL-C, CTRL-V"). I was permanently humbled after that experience.

Thanks especially to Richard Harang, Ph.D., and Alexander Kott, Ph.D., and their colleagues at the Army Research Laboratory's Network Science Division for providing this data and affording me the opportunity to leap headlong into their area of expertise and make what I hope is the first of many more fruitful collaborations between our

organizations. Your hospitality during the week we spent at ARL was truly the most eye-opening and educational site visit I have ever experienced.

I. INTRODUCTION

A. BACKGROUND

Security in cyberspace, or *cyber security,* is an area of growing National and Department of Defense (DoD) interest. The formation of the US CYBERCOM, the Navy's TENTH FLEET and the merging of the Office of the Chief of Naval Operations (OPNAV) Naval Intelligence (N2) and Communications (N6) directorates mark the recognition of cyber warfare as a primary warfare area, as well as the *National Strategy for Cyberspace.*

Applications of Operations Research to Cyber Security is a growing area, acknowledged both by the Cyber community of practice and the OR community. In March of 2011 the Military Operations Research Society (MORS) held a special meeting titled *Mission Assurance: Analysis for Cyber Operations* for which seven working groups convened to discuss issues linking OR to cyber studies. Among the panels' findings were the following opportunities for the application of OR capabilities:

- Force-on-force analysis that accurately accounts for cyber effects and actions.
- Statistical process control techniques to enhance situational awareness and threat awareness.
- Design of experiments methodologies to help assess rapidly fielded equipment and systems.
- Application of neural networks to help detect anomalies and hostile activity.
- Decision analysis tools and techniques to facilitate response to attacks.
- Optimization/matching techniques to address requirements prioritization.
- Manpower analysis tools and methodologies to assist with those issues [1].

The Synthesis Working Group closed their findings with the following recommendation:

> The Synthesis Working Group observes that the analysis communities across the Services need to make applying these and other methods in our

> toolkits to the Cyber arena a priority. For this to be successful, however, it is critical that there be an associated "pull" from the Cyber community itself for this kind of help. We emphasize that the Leaders' roles in both communities are key! [1]

An example of this leadership "pull" was demonstrated in large part at the 80th MORS Symposium where General William Shelton, Commander, Air Force Space Command served as the keynote speaker and 28 working groups addressed issues related to OR applications to cyber operations [2]. Also, in September of 2011, the Naval Postgraduate School established the Cyber Academic Group and introduced a Cyber Systems and Operations Curriculum to address the cyber topics relevant to DoD leaders and operators [3]. Of the many facets of analysis in cyberspace, we focus on a specific problem—intrusion detection on a trusted network. In particular, we investigate the utility of a particular dataset intended to provoke innovative research in the detection of attacks on a U.S. Intelligence Community unclassified network.

The transition in US Forces to increasing lightness and pin-point lethality has come at a cost of increased reliance on computers and the connections between them. We loosely call this "the network." Because data—both operational and planning—transits this network, it is a ripe target for adversaries to attempt to either extract valuable information or deny the use of it. A characteristic of conflict in cyberspace or cyber conflict is the speed at which it moves. This rapidly changing landscape means that analysts rarely have the opportunity to ask higher-level, policy questions like, "What is the best statistical tool to use against cyber data generally." The acknowledgment that the operators are overtasked coupled with the sense that better analytical efficiency is possible has led directly to this work.

In order to ensure the security of the network, operators—specifically, the U.S. Army—collect data from their analysis, but they also generate data sets for experimental purposes. The reason that artificial datasets are required is twofold; first, real data is, for lack of better words, real. This means that it contains security and personally sensitive information which organizations are reluctant to share. Secondly, statistical analysis

requires knowledge of “ground truth;” in real-world data, we do not know which events are “bad,” how many “bad” events there are, just how many we think are “bad.”

Both real and experimentally generated datasets can be quite large, in some cases exceeding one gigabyte for less than 15 minutes of captured network traffic. The captured data flows back to headquarters on the same network “pipes” that the traffic itself flows on; therefore, the act of sending network data in some measure reduces the network’s effectiveness. This creates a challenging analytic problem in that the data used to maintain the health of the network itself reduces the capability of the network. The amount and type of data collected needs to be carefully chosen to balance between the desires to keep the network “fast” and “secure.” A perfectly “fast” network that has no diagnostic data would not be safe; a perfectly “safe” network that is so clogged with metadata as to be unreasonably slow is of no use, either. Our work seeks to find better predictors in the provided data to find a balance between security and speed.

Data that is currently collected on cyber-attack discoveries is used forensically by the Army and others to build and improve tools for detection and prevention of similar attacks, but work remains to be done to determine to what extent the same data can be used to detect larger trends and relationships, or whether any particularly useful signals exist in the data.

Software developed to analyze network traffic for security purposes can be developed to maximize sensitivity (i.e., better threat detection without regard to false positive rate), or specificity (i.e., better overall classification accuracy). Tools for threat detection can be categorized by whether they detect threats based on specific signature characteristics or based on more general statistical characteristics. Signature-based detection tools detect known threats by seeking exact matches within network traffic to very specific rule sets developed from characteristics seen in previous attack attempts. These criteria can take many forms, but they usually require deep packet inspection of the information “payload” contained within a network packet. This type of “fingerprint” analysis differs greatly from the more general data-collection and probabilistic analysis

associated with the use of statistical methods, which attempt to categorize traffic into categories based on more surface level data in order to discern “good” traffic from the “bad.”

Statistical tools can be further categorized by the type of learning methods they employ. Supervised methods are classification methods that rely on inputs of prior, correctly-labeled good and bad traffic in order to make predictions of the nature of future traffic. Regression models, kernel function methods, trees, and neural networks are examples of supervised methods. Unsupervised learning, on the other hand, occurs without a previously labeled set of data. These methods generally attempt to detect anomalous traffic based on its relationship to other traffic, or its “outlierness.” These methods compare traffic data based on their attributes and attempts to establish clusters in order to discern normal from anomalous traffic. This principle relies on good traffic falling into relatively “normal” categories, and on bad traffic having distinct, measurable characteristics. Finally, while anomaly detection is generally done using unsupervised learning methods like clustering, there exist supervised methods of anomaly detection as well.

Another distinction among detection tools is whether they operate on-line or off-line. This refers to whether a particular tool can inspect the traffic and make an evaluation in real or near-real time based only on what it has seen up to that point, or whether it must wait to analyze an entire interval of data either because the computation takes too much time and/or memory to conduct in real time, or because the evaluation is only meaningful in the context of the entire interval of data.

Furthermore, anomaly detection tools of both types can also be developed to integrate alerts to general network performance characteristics, such as data throughput at a specific node (a server, or a client computer). These can be useful in detecting the symptoms of an attack exploitation which may otherwise seem quite “normal” from the perspective of individual packet inspection or IP address statistics. This signal integration could be particularly helpful in detecting so-called “zero-day” attacks, i.e., those that rely on previously unknown vulnerabilities. For example, W32.stuxnet

contained 4 “zero-day” attacks [4]. For these novel attacks, analysts generally rely on more tailored tools and techniques—often developed and personalized by the analysts themselves—to apply what is best described as the art of network security analysis.

Finally, the establishment and maintenance of what defines “normal” network activity is the most difficult part of anomaly detection. The possible characterizations of malicious activity these tools must detect largely overlap the spectrum of normal network activity characteristics [5]. Malicious activity could present as seemingly “normal” e-mail phishing attempts that lure users to malicious web-sites or into fraud scams. Another “normal” presentation of malicious activity is port scanning, in which an attacker probes servers and computers to find open, unused ports through which they can enter the network undetected. At the other end of the spectrum could be specifically directed cyber-attacks which may include objectives such as denial of service (DOS) (such as those used by Russia in their conflict with Georgia)[6], attempted exfiltration of valuable proprietary, or national security related data from a targeted network location.

B. OUR ANALYSIS

For our study, the Army Research Laboratory Network Science Division, located in Adelphi, Maryland, and Aberdeen Proving Grounds has provided a labeled dataset derived from the Skaion Corporation’s Advanced Research and Development Agency (ARDA) Testbed. The primary data in the set is consists of three different scenarios, each of which includes simultaneous network packet captures from three different nodes, or sensors, in the simulated network. For the two scenarios we analyze, these capture files record a 15 and a 30 minute window of network traffic from the three different locations in the form of packets, which are the basic unit of network communication. From these packets, characteristics such as source and destination IP addresses, source and destination port, network protocols used, packet size in bytes, etc. can be analyzed. These characteristics are described in detail in subsequent chapters. To illustrate the problems with collecting network data, the 45 minutes of data collected from the three different sensor locations has a size of approximately 4 gigabytes (uncompressed).

Included with these packet capture files are attack logs from four signature- based network intrusion detection systems as well as IP address keys that provide the "labeling," or ground truth as to the roles of each IP address. In other words, the point of origination of each network packet is classified in the file as an attacker, victim, background attacker, background scanner, server or user. This $1.3 million network emulation project was contracted by the Defense Advanced Research Projects Agency (DARPA) in 2003 [7] to simulate realistic attacks on a notional U.S. intelligence community (IC) network for the purpose of fostering "innovative research on the protection of highly-sensitive assets such as the IC's own networks and systems [8]." The data to be analyzed consists of network packet capture files (PCAP), alert logs from a variety of intrusion detection systems (IDS), and an IP key that provides ground truth as to the nature of an IP address' role in the network (i.e., attacker, victim, or benign). While our analysis focuses on a 2003 dataset, the methods and techniques we propose are broadly applicable.

C. LITERATURE REVIEW OF PREVIOUS WORK

For high-level discussions of the cyber problem and its importance, we recommend the *Cyberspace Policy Review* [9]. The DoD Lexicon is given in Joint Publication 6-0 *Joint Communication System* [10]. For an in-depth description of different types of cyber-attacks, see *The Art of Computer Virus Research and Defense* [11], and for a DoD approved discussion of open problems and their import, see the JASON report *The Science of Cyber-Security* [5].

For the precise application of statistical methods to network data analysis, we rely on Yale University's Yun Wang [12]. Wang provides a guide to statistical methods in the field of network security by taking a comprehensive approach to network security history, term definitions, data mining and modeling techniques, and classification methods. Most germane to this study are chapters 9 and 12, which examine supervised modeling and classification methods suitable for prediction as well as methods of comparing and rating those models.

Gogoi, Bhattacharyya, Borah and Kalita outline eight supervised and three unsupervised approaches to outlier detection in network security [13]. Consistent with

our observations, they describe that the primary difficulties with anomaly-based network intrusion detection are in the assumptions that:

> 1. The majority of the network connections are normal traffic. Only a small amount of traffic is malicious.
>
> 2. Attack traffic is statistically different from benign, i.e., 'normal' traffic.
>
> However, in a real-world network scenario, these assumptions may not always be true. For example, when dealing with DDoS (distributed denial of service) or bursty attack detection in computer networks, the anomalous traffic is actually more frequent than the normal traffic. [13]

The first assumption is generally reasonable. The second assumption may hold when the attacker is a disorganized, or loosely organized criminal activity. However, a state- or state-sponsored effort should be expected to be sophisticated enough to "cover their tracks."

Mahoney and Chan [14] examine the 1999 DARPA/Lincoln Laboratory off-line evaluation data set (IDEVAL [1] – [3]) and compare the results of six basic anomaly detection (AD) systems against 244 labeled instances of 58 different attacks present in the data set. The six systems are chosen more for their collective utility with regard to different types of Internet traffic than for their particular detection abilities. The purpose of their analysis is to confirm whether mixing real traffic in with the IDEVAL data set improved the AD systems' learning.

The study finds that mixing improves the detection percentages of all six AD systems. This improvement is attributed to the difference in prevalence of "poorly-behaved artifacts with power law attributes" [14] in the IDEVAL set (low prevalence) and in the real traffic (high prevalence). The higher prevalence of these poorly behaved artifacts in the real traffic had the effect of removing them as artifacts, even without having to define what the suspected artifacts were. The results demonstrate that when using mixed data, the detection performances of the very basic AD systems chosen for the study are raised to levels comparable to the best-performing systems used in the 1999 evaluation.

Yamanishi and Takeuchi, 2002, demonstrate that an online unsupervised outlier detection program called SmartSifter can be useful in detecting abnormal activity in both intrusion detection and health insurance pathology contexts. SmartSifter performed well in a supervised mode at the 1999 KDD Cup Machine Learning competition where it detected 55% of 1,687 intrusions out of a set of 458,078 network accesses in its top 1% of ranked data and 82% in its top 5% of ranked data [15]. The novel features of SmartSifter are threefold: adaptability to non-stationary sources of data, its rank score has a clear statistical and information-theoretic meaning, and it is computationally inexpensive and it can handle both categorical and continuous variables [15].

Tavallaee, Bagheri, Lu, and Ghorbani [16] conduct a critical analysis of the KDD-99 dataset mentioned above. They discuss its design shortcomings and propose a new dataset, in which they mitigate two main issues they find with the original dataset; namely, that there are too many duplicated records in both the training and test sets, and that results among different methods are difficult to distinguish due to a narrow margin of difference in performance. Their corrected dataset is a subset of the original data that eliminates duplicated records and adjusts the ratio of records based on a relative level of classification difficulty such that the numbers of records of each difficulty level are inversely proportional to the numbers found in the original set. Their results show both a lower overall accuracy among their chosen models as compared to the original dataset, (which shows the tendency of the original set to overestimate performance) as well as more varied performance among models. Finally, they acknowledge that although the KDD-99 data is problematic, it remains a benchmark due to the lack of availability of publicly available datasets.

Wang and Kim, 2008, attack the problem of insufficient and unavailable training data (and the subsequent unsuitability of regression models and neural networks) in network security problems. Their supervised approach uses a bootstrap resampling method to develop a probability model for use when limited or no abnormal information is available in training data [17].

D. OBJECTIVES

This study investigates methods of variable selection, data coding and modeling in the context of the Skaion data set. The intent is to analyze selected, likely successful data mining and statistical methods and report on their utility with respect to detection probabilities and false positive rates in the Skaion data set. Our research questions are: "What are effective methods of data collection and organization in order to evaluate the performance of logistic regression in detection of cyber-attacks against the Skaion dataset, and what general conclusions may be drawn for statistical techniques for use in network detection?"

This analysis provides the Army Research Laboratory specifically and the Department of Defense generally with insight into the Skaion data set, as well as foundational work for further Operations Research cyber security analysis efforts.

E. SCOPE, LIMITATIONS AND ASSUMPTIONS

The Skaion data set is composed of five attack vignettes and also has multiple releases. This study focuses on attack vignettes 4s1 and 4s3 from release 4a. Attack 4s1 is a basic Common Gateway Interface (CGI) Overflow attempt. A CGI is a commonly used application framework for programs like Internet shopping carts and website search engines and they can be hijacked by code intended to disable their ability to allow limited access to the information and program function of the host. Instead of a shopping cart, Skaion uses a petition website linked to an SQL database for storing the signers' data. The attacker uses a malicious code to overflow the CGI application's buffer, allowing full access to the information in the database. Scenario 4s3 is a variation of the same attack that attempts to mask its true point of origin of the main attacker by sending out decoy requests that trigger the same alerts from the *Snort* intrusion detection systems (IDS). Both attack scenarios also include the background attacks and scanners that are inherent in the Skaion test bed design.

The study focuses analysis from an IP address-centric approach and investigates methods to detect whether, within the time frame of the data set, a particular IP address is malicious. When examining packets, the response variable is determined by the ground-

truth classification of the source IP address as either "ATTACKER," "BACKGROUND ATTACKER" or "BACKGROUND SCANNER." The primary attacker of interest and the background attackers and scanners comprise the subset of malicious IP addresses. Scanners are IP addresses that run programs designed to test which ports are open on a host or server. When a port is open or active, the service that is running on that port may be used to exploit network security vulnerabilities [12].

We are primarily concerned with the statistical aspects of the problem, and guiding investments at a mid- to high- level. We do not attempt to conduct a detailed forensic analysis of the Skaion data set and we have necessarily abstracted away many of the more detailed aspects of computer coding, etc. The primary reason for this is to avoid incorrect inferences relative to the context of the data and the precise nature of the relationships among the classifiers.

Finally, the study does not provide an in-depth analysis of the strengths and weaknesses of the IDS tools that provide alert responses in the data set. Rather, the responses serve as classifiers and their results are compared as subsets of the overall data set to draw out the significant characteristics of the types of traffic that generate alerts at varying levels of true and false positive and negative.

F. COURSE OF STUDY

The outline of the thesis is as follows: In Chapter II, we cover the nature of the data set, providing a comparison of the two attack scenarios and a description of the Skaion ARDA Testbed Design. In Chapter III, we describe the nature of the variables collected, the methodologies used. In Chapter IV, we discuss the results and analysis of the chosen methods against both the Skaion dataset and another representative dataset. Finally, in Chapter III, we cover the conclusions and recommendations for further research.

II. DATA

A. DATA SET BACKGROUND

1. Skaion ARDA Testbed Design

Skaion chose to design its test bed after the Open Source Information System (OSIS), which in 2003 was the U.S. intelligence community's unclassified open source intelligence network. The OSIS network and its content were de-coupled in 2006 and replaced with DNI-U and Intelink-U, respectively [18].

This choice of OSIS was guided by Skaion's contract requirement "to simulate realistic attacks on a notional U.S. intelligence community (IC) network for the purpose of fostering "innovative research on the protection of highly-sensitive assets such as the IC's own networks and systems" [7]. Skaion did not conduct attacks against the actual network, but instead constructed a simulated model of the OSIS network and used network emulation algorithms to build realistic network traffic on a test bed of approximately 20 physical machines. The test bed includes real (i.e., physical) hosts (servers and clients that maintain websites), network infrastructure components (routers, hubs and firewalls that control network access and flow), and several hundred virtual hosts simulated by traffic generation machines using Skaion's proprietary network emulation algorithms. The simulated OSIS network is connected to a model Internet, which is simulated by two other traffic generation (emulation) machines. The structure of the complete OSIS-Internet model is given in Figure 1.

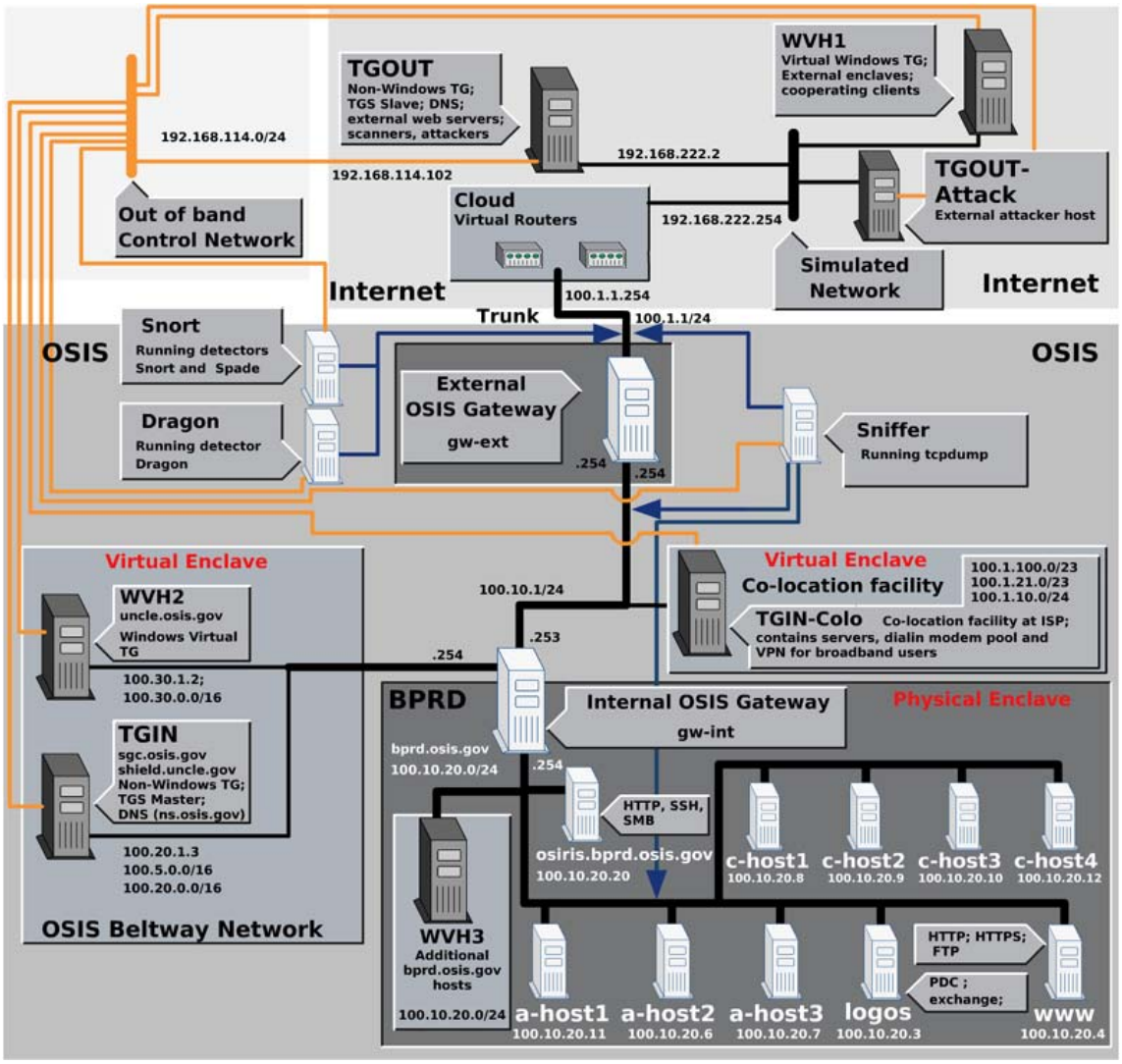

Figure 1. Architecture of Skaion's test bed design. Sensor locations are indicated by the blue arrows. The machines labled "Sniffer" run a program called tcpdump, which collects and inspects each packet passing through the indicated point in the network. The machines labeled "Snort" and "Dragon" are running intrusion detection system (IDS) programs by those names that do some automated analysis of the packets passing through that point in the network. Alerts are generated based on pre-determined rule sets (i.e., correlation between a packet's originating IP address and a "hot list" of known malicious IP addresses). (Image from unpublished documentation included within the Skaion Data Set).

The background traffic generated by the Skaion Corporation is representative of the Non-secure Internet Protocol Network (NIPRNET) traffic at the Air Force Research Laboratory (AFRL) in Rome, NY. NIPRNET is the common infrastructure for unclassified general administrative and training data and is the closest analogue in the DoD to civilian business IT Enterprise systems. The traffic is anonymized by transforming IP and Ethernet addresses and replacing IP and Transmission Control Protocol (TCP) payloads with simple checksums. Anonymization is required to protect the privacy and personal information of the AFRL users. The data collection portion of the scenarios we analyze was conducted in January and February of 2004 and consists of a 15 to 30 minute capture (scenario dependent) from three different network sensors. These sensors are placed at different locations in the network and operate like a listening/recording device. These six separate 15–30 minute (~2 hours 15 minutes) of packet captures sum to approximately 4 gigabytes of total data. To put this in context, this equates to about 8 hours of high definition recorded video, or about 1,000 mp3 music files.

The entire project conducted more than the two attack scenarios that we address, but only one of these was included in the data set provided by ARL's Network Science Division. Because this attack scenario focused on an e-mail phishing attack, as described in Chapter I, its relevance to the other two data sets was minimal and we chose not to include it in our analysis. Anonymization of the data is inherent in the design of the network emulation, since it does not retain any actual user activity. This is an important feature due to the sensitivity of what otherwise could include personally identifiable, and exploitable, information.

The Skaion dataset's companion documentation outlines the following limitations that exist due to anonymization of the network traffic:

- Cannot characterize the relative likelihoods of types of networks that could be used to probe another network
- Cannot distinguish between probes, attacks, and non-malicious but broken traffic
- Cannot tell when a non-standard protocol is run over a standard port (NFS over SMPT, or backdoors run over port 80).

2. Attack Scenarios

The dataset provided by ARL's Network Science Division was a partial dataset that included three attack scenarios. We focused on two in particular, using the first scenario (4a4s1) as a training dataset, and the second scenario (4a4s3) as a test dataset.

a. Dataset 4a4s1: Basic CGI-Overflow Attack

The basic attack in this dataset is an attempt to conduct a buffer overflow attack against a Common Gateway Interface (CGI) script. The CGI script allows users to log on and digitally sign a petition, the signatures of which are stored in an SQL database. The script connects with the database and uses the buffer overflow to query all the information in the database, gaining access to what should be protected information.

b. Dataset 4a4s3: CGI-Overflow with Chaff

Attack scenario 4a4s3 is similar to 4a4s1 except that the attack is preceded by sending sporadic requests that produce the same Snort IDS alerts as the actual attack, thus masking the true attempt.

3. Summary Statistics from Scenarios 4a4s1 and 4a4s3

Table 1 illustrates some of the major differences between the two attack scenarios. The most notable difference is in the distribution of malicious activity, but the difference is not as clear-cut as one might initially suspect. Because the intent of 4a4s3 is to create more chaff traffic to mask the intrusion, one might think this means more overall malicious activity, or at least a higher density of malicious activity.

While 4a4s3 does have 47% more malicious IP addresses active within a capture period that is half as long as 4a4s1, the overall numbers of packets and kilobytes associated with these malicious IP addresses drops by 80% and 93%, respectively, from 4a4s1 to 4a4s3. While it makes sense that the 50% decrease in time of capture should necessarily decrease the amount of overall malicious traffic, this does not account for the fact that the overall density of malicious traffic, represented by the ratio of malicious activity to total activity, drops from 4a4s1 to 4a4s3 by a factor of 10.

Table 1. This table lists some of the key differences between the two attack scenarios. 4a4s1 captures twice as much time and approximately 30% more data, but generates less than half the number of Diagnosis Events as 4a4s3, which has more attackers (25 versus 17 in 4a4s1), but is ten times less dense with activity from malicious IP addresses.

Differences between 4s1/4s3		
	4s1	**4s3**
Capture Duration (min)	25	13
Traffic Totals		
Kilobytes	904,634	592,336
Packets	1,240,299	892,219
IP Addresses	**1974**	**1878**
Benign IP Addresses	1957	1853
Malicious IP Addresses	17	25
Percent Malicious IP Addresses	0.86%	1.33%
Percent Malicious IP Packets	0.09%	0.02%
Percent Malicious IP Kilobytes	0.10%	0.01%
Total Conversations	**23,443**	**16,256**
Physical Conversation	44	49
IP Conversation	2,423	2,260
TCP Conversation	20,165	13,041
UDP Conversation	811	906
All Diagnosis Events	**8,507**	**26,884**
Application Layer Events	**2,795**	**2,527**
DNS Server Slow Response	11	1
DNS Host or Domain Does Not Exist	1,132	1,270
DNS Server Error	1,250	836
SMTP Server Slow Response	28	21
POP3 Server Slow Response	31	1
POP3 Server Returned Error	299	41
FTP Server Slow Response	8	345
FTP Server Returned Error	1	3
HTTP Client Error	2	2
HTTP Suspicious Conversation	2	1
HTTP Server Slow Response	31	6
Transport Layer Events	**5,510**	**24,130**
TCP Connection Refused	588	12,065
TCP Connection Retry	1,370	696
TCP Retransmission	102	6,796
Illegal TCP Checksum	52	68
TCP Slow Response	2,813	3,707
TCP Duplicated Acknowledgement	585	798
Network Layer Events	**202**	**227**
Illegal IP Checksum	202	215
General network layer fault alarm	0	12

Furthermore, although scenario 4a4s3 is half as long and less dense with traffic from malicious IP addresses by a factor of 10, the traffic those IP addresses are sending results in more than twice the number of diagnosis events as compared to 4a4s1 (see Table 1). Diagnosis Events are system and security alerts that occur in the normal interaction between systems and are encoded in the packet details. Many Diagnosis Events are benign indications of system performance issues and incorrect settings, but some can be direct indications of malicious activity. For example, the main attacker in both scenarios is the only IP address to cause the alert below "HTTP Suspicious Conversation," which occurred because it sent non-HTTP encoded traffic through an HTTP port.

B. ACCESSING THE DATA

Pre-processing the data for analysis represented a significant portion of our analytic effort, and took approximately 400 analyst-man hours (or 10 man-weeks) to accomplish. More than anything else, our experience with transforming the data to a useful analytic form highlights the difficulties with applying statistical methods to computer network data, as the formats are typically not amicable to analysis. Much of what follows was determined by working with software that was poorly documented—when it was documented at all—therefore much of this work was performed by trial-and-error. For some Visual Basic (VBA) code used in preprocessing data, see Appendix A.

Because of the difficulties involved with anonymization of data and generalization of network vulnerabilities, there are few publicly available network security datasets. Those that are available can be placed in two basic categories. The first is, like the Skaion dataset, largely unprocessed packet capture data accompanied by some kind of key which indicates attacks or attackers. The second category comes with some amount of pre-processing already complete. This type of dataset allows the analyst to bypass the major data storage and processing overhead and simply test models assuming the variables included in the dataset are sufficient.

A good example of the second type of data set is the 1999 KDD-Cup data set. This is an older data set that was developed for The Third International Knowledge Discovery and Data Mining Tools Competition, and it is considered an industry standard

for testing statistical methods against network security threats [19]. The main difference between the KDD-cup and Skaion data sets is that the KDD-cup is presented as collections of comma separated values (.csv) files in which each row represents a connection between two IP addresses. Each row is labeled as either "normal" or as one of 24 attack types, and each connection is described by 41 predictor variable columns. The other main difference between the data sets is that KDD-cup is far more attack dense than Skaion. For a more in-depth description of the KDD-cup dataset, see Appendix B.

1. Unpacking, Opening and Exporting Data from Packet Capture Files

Skaion captured their packet data using freeware called TCPDump, stored the files using the .dmp ("dump") file extension, and compressed that file into a .tar.gz file. The .tar extension name is derivative of a holdover term, tape archive, which is an outmoded method of data capture and storage. The .gz extension means GNU (which is a recursive acronym that stands for Gnu's Not Unix) Zipped Archive file (.gz). This compression method is optimized for packet data and is easily decompressed in a Unix environment or with open source software on a non-Unix machine. We used a program called 7-Zip to decompress and unpack the .tar.gz file.

Our choice of software for inspection and pre-processing the capture files was also an open source program called Capsa from Colasoft®. This program is available in both free version and enterprise versions. Because we were only viewing and processing already captured data and not setting up a network monitoring sensor ourselves, the freeware version was sufficient.

Because Capsa could not read the .dmp file format of Skaion's capture files, we used another popular network protocol analyzing software package called Wireshark (Development Version 1.7.0) to open the .dmp file and save it as a Wireshark capture file (.pcap). Once open, Capsa was used to accomplish the following:

1. Combine all three files and collect summary statistics on the entire capture file from all three sensor locations,
2. Open each sensor capture file individually for pre-processing and export of data sorted by IP address,
3. Export all IP Statistics (.csv),

4. Export all Diagnosis Events (.csv),

5. Export all Diagnosis Items (.csv).

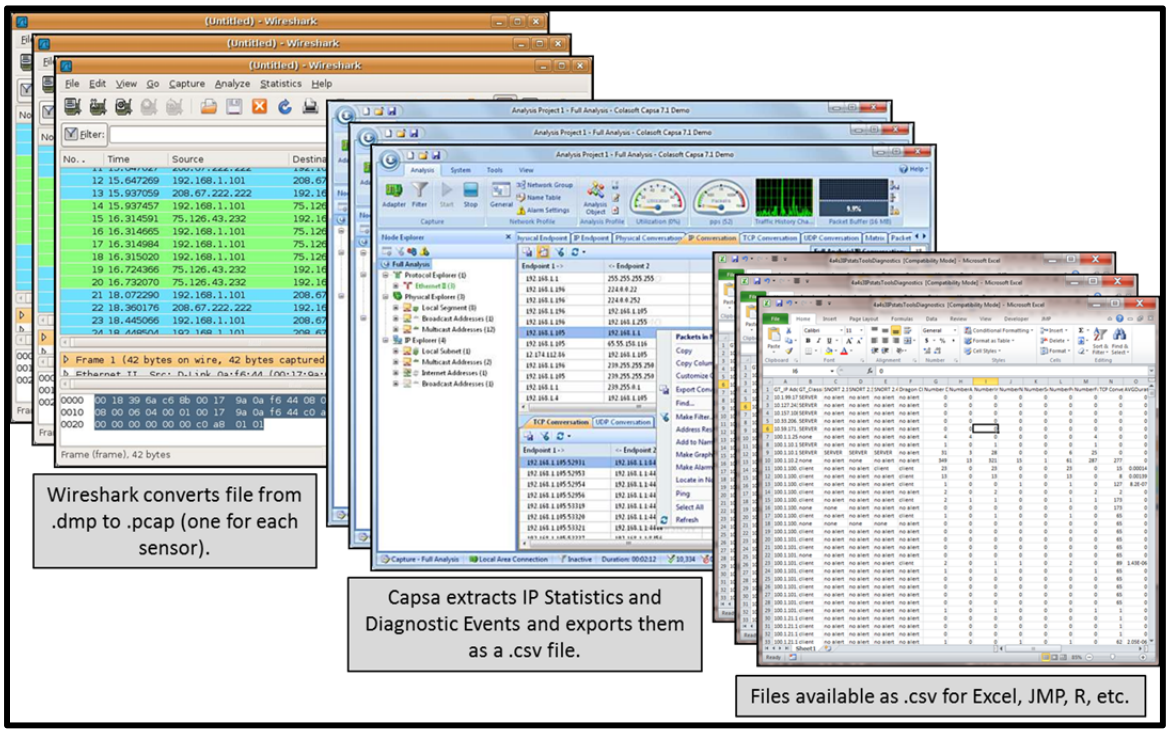

Figure 2. The datasets were manipulated exclusively in a Microsoft Windows® environment, using Wireshark® to convert the capture file extension so Capsa® could be used to extract statistics. Microsoft Excel® was used to organize the data in a format exportable to SAS's JMP® statistical software.

2. Processing the Exported Data

Along with the raw packet data, Skaion included an IP Address Key that identified the majority of IP Addresses in the file as attackers, victims, client servers, background attackers, background scanners or as nothing at all (these amounted to relatively few, and thus were assumed to be the source of additional benign background traffic). These IP Keys were used to "label" the data set according to the roles the IP Address plays in each scenario.

Another file, external to both Capsa and the provided capture files, was a key providing descriptions and the event code number to each diagnosis event. These code numbers can be used as dependent variable labels and their columns would indicate how

many of each diagnosis event could be attributed to each IP Address. These codes can also be summed by IP or conversation based on the priority of the event (Alarm, Notice or Information), or by the nature of the event (Security, Performance or Fault). The file was not all-inclusive of every diagnosis event possible, but only the ones seen in the two attack scenarios in the data set. This amounted to 18 events.

Once the pertinent data was exported from Capsa, the following steps were completed in Microsoft Excel® (See Appendix A. for macro code):

- For each sensor location (COLO, BPRD, TRUNK) and each scenario (4a4s1 and 4a4s3) combine IP Statistics.csv, Diagnosis Events.csv, Error Key.csv, and IP Key.csv,
- Rename each tab using the following generic names: IP Stats, Diagnosis Events, Error Key, ip-key
- Save file as 4a4sX XXXX IP Stats.xlsx (XXXX = COLO, BPRD or TRUNK),
- Import macro module (consisting of the macros in Section C. below),
- Run AddErrorClassColumns() macro,
- Run DeleteCountryRows() macro,
- Delete Column BC, then scroll down and delete all excess rows (those without IP addresses in Column A) at the bottom of the active worksheet,
- Run FillErrorClassCols() macro,
- Rename new tab “4A4SX XXX Final.xlsx,”
- Run DeleteExtraDataTags() macro,
- Run AddClassCols() macro,
- Delete extra rows at the bottom of the worksheet,
- Close active workbook and repeat for all locations and scenarios.

C. VARIABLES

1. Dependent Variables

We identify the following variables as dependent, as they are derived from the ground truth IP key provided by Skaion. Their use as target variables can be enhanced by recoding on a binary scale (malicious IP or benign IP) or an ordinal scale (ranking of IP addresses by their relative threat levels). There is only one main attacker IP address in each data set, and there are 23 and 24 background attackers and scanners in 4a4s1 and 4a4s3, respectively.

These dependent variables can be explored from two different perspectives in this data set. One way to organize the data is by IP address, which yields one row of data for each of the 1,976 or 1,883 IP addresses in each attack scenario. The other method is from the perspective of connections, or conversations, between two different IP addresses. We address the advantages, disadvantages and model performance from both perspectives.

Table 2. IP Key and ground truth of dependent variables. IP classifications can be ranked categorically by classifier, by ordinal threat level, or by a binary flag indicating malicious or benign. These recoded classifiers help facilitate multiple modeling methods.

Key Classification	Explanation	Binary Recoding	Ordinal Recoding	No. 4a4s1	No. 4a4s3
ATTACKER	Primary attacker of interest (only one in each file)	MALICIOUS (1)	4	1	1
BACKGROUND ATTACKER	Attackers inherent in the Skaion ARDA test bed design, provided for realism	MALICIOUS (1)	3	4	5
BACKGROUND SCANNER	Scanners inherent in the Skaion ARDA test bed design, provided for realism	MALICIOUS (1)	2	21	21
CHAFF	Attackers provided in 4a4s3which emulate characteristics of the actual attacker's behavior in order to mask the actual attacker's origin/identity	BENIGN (0)	1	0	5
SERVER	Neutral role description	BENIGN (0)	0	1596	1509
CLIENT	Neutral role description	BENIGN (0)	0	289	277
VICTIM	Target of ATTACKER	BENIGN (0)	0	1	1
[Blanks]	Unclassified (benign) background traffic	BENIGN (0)	0	65	64
TOTAL				1976	1883

2. Predictor Variables

We divide the dependent variables into three categories based on their sources. The first set comes from comma separated value (.csv) files within the data set that give IP address role predictions of four different intrusion detection (IDS) tools. These predictions are similar in format to the dependent variables described in Table 1. We refer to these variables as **Tool Predictions**. The IDS tools used by Skaion were Dragon, and three different versions of Snort (versions 1.2.1, 2.3.2, and 2.4.1). The basic differences between the rule sets for these tools are described in Appendix X, but the scope of this thesis does not include an in-depth discussion or analysis of these rule sets. Rather, we utilize these tools as a radar operator may view four different radar systems designed to detect different types of threats or operate better in different types of environments. Leveraging the responses from these tools allows us to explore methods of comparing IDS performance and determining how to best utilize the valuable information provided by these tools and their highly contextual rule sets.

The second set of variables, which we will refer to as **Diagnostic Events**, are summations of the types and severity levels of alarms generated by the Capsa software we used to access the packet data. Like the **Tool Predictions**, the coding and computing overhead associated with generating the responses is already completed by the software, so collecting the data for these variables is a simple process that consists of exporting the event log using the software's graphics user interface (GUI) and associating the alarms with IP addresses in the main data set. The categories and codes associated with the different alarms are described in Table 2.

The third set of variables, which we refer to as **Conversation Statistics**, reflect the nature of the data transfer that occurs between two IP addresses. These variables are listed and described in Table 3.

The following predictor variables represent aggregated network statistics for each individual IP address collected over approximately a 15 minute window. The data was analyzed and aggregated using a combination of Wireshark and Capsa. These programs decompress, decode and organize data stored in packet form so that it can be reorganized

and output in .csv or .xls file formats for statistical analysis. The variables described in Table 3 result from organizing the packet statistics by IP Address. They describe the behavior of each IP address in its role as an originator or end receiver of data.

Table 3. Predictor variables. Generated from data encoded in packets, these variables provide easily aggregated numerical statistics describing the network behavior of a particular IP address over a short time period (continued on next page).

PREDICTOR VARIABLES

INTRUSION DETECTION SYSTEM (IDS) PREDICTION CATEGORY	
SNORT 2.1.2	IP role prediction generated by the SNORT IDS, version 2.1.2 (see Table 2 for values)
SNORT 2.3.2	IP role prediction generated by the SNORT IDS, version 2.3.2 (see Table 2 for values)
SNORT 2.4.1	IP role prediction generated by the SNORT IDS, version 2.4.1 (see Table 2 for values)
Dragon	IP role prediction generated by the Dragon IDS (see Table 2 for values)
IP CONVERSATION STATISTICS CATEGORY	
Name	IP Address
Bytes	Number of total bytes (in kilobytes) sent or received by the IP address
Packets	Number of total packets sent or received by the IP address
Bytes Received	Number of bytes received by the IP address (in megabytes)
Packets Received	Number of packets received by the IP address
Bytes Sent	Number of bytes sent by the IP address (in megabytes)
Packets Sent	Number of packets sent by the IP address
Protocol	The type of information being sent (TCP, HTTP, etc.)
Source Port	The port used by the sender (i.e., Port 80 for HTTP traffic)
Destination Port	The port used by the recipient
DIAGNOSTIC EVENTS	
NUMBER ALARMS	**Total number of alerts with a priority level "alarm" generated by the IP Address**
HTTP Suspicious Conversation	A HTTP 80/TCP connection contains non-HTTP traffic.
TCP Duplicated Acknowledgement	Number of times a TCP ACK packet is captured more than 3 times. This is a performance fault generated either because of a mis-sequence or retransmission of a lost packet.
NUMBER NOTICES	**Total number of Total number of alerts with a priority level "notice" generated by the IP Address**
DNS Server Error	A DNS Server returns an error except "host name or domain name which client requests is inexistent," which means the client requests or domain fail to return.
SMTP Server Slow	The average server response time is equal to or higher than the Slow

Response	Response Time threshold.
HTTP Server Slow Response	The average server response time is equal to or higher than the Slow Response Time threshold.
NUMBER INFORMATIONS	**Total number of Total number of alerts with a priority level "information" generated by the IP Address**
TCP Connections Refused	Number of times a client's initial TCP connection has been rejected by the target host. This fault is generated either because a client is requesting a service a host does not offer or the server is overloaded and cannot establish new connections.
Repeated Attempts to Establish TCP Connection	Number faults generated due to a client making repeated unsuccessful attempts to establish a TCP connection. This could be due to a firewall blocking the synch (SYN) request packet from the server to the client.
TCP Retransmissions	Number of times an IP address resubmits a packet. This is a network performance problem due to network or receiver overload, or a network delay causing acknowledgement packets to be sent slower than the incoming packets.
TCP Checksum Error	Number of times a TCP header contains an erroneous checksum. This fault could be due to a faulty device in the network or, if all local packet checksums are invalid, it is due to an incorrect setting (checksum offload function enabled).
TCP Slow Response	The number of times an IP address generates a fault due to the average response time of TCP connections exceeds the average response time of the connection plus the Slow ACK Time threshold.
DNS Host of Domain Does Not Exist	Host name or Domain name which client requests does not exist.
FTP Server Slow Response	The average server response time is equal to or higher than the Slow Response Time threshold.
FTP Server Returned Error	A FTP connection or request is rejected by a FTP server after a TCP connection has already been established.
Illegal IP Checksums	Number of times a packet has an error in the checksum of an IP header. The checksum value is calculated by the sender and written to the packet, and then recalculated from the received packet by the receiver. It indicates an error if the two values are different. This is typically caused by a faulty device in the network (Colasoft help).
HTTP Client Error	HTTP server returns a 4xx error code other than 404 (Request Not Found) to indicate a client error. The client's request is incomplete or forbidden.

Table 3. Predictor variables (Continued from previous page).

THIS PAGE INTENTIONALLY LEFT BLANK

III. METHODOLOGY

The regression results in Chapters III and IV are generated by default models in SAS's statistics package, commonly referred to as "JMP" (Pro Version 10), implemented on a Gateway ZX6971 with an Intel® Pentium® 260GHz processor running the 64-bit Windows 7 Home Premium operating system with 4 GB RAM. For details reference the online JMP documentation [20].

During data exploration, we experimented with several methods including neural networks, classification and regression trees, and clustering. Given the same types of variables, these methods either returned results that were less accurate than logistic regression or achieved better predictive results through over-fitting the data.

A. EXPLORING DEPENDENT VARIABLE PERSPECTIVES

The data created by Skaion is "labeled" according to the roles played by individual IP addresses (attacker, background attacker, background scanner, victim, server, etc.). In other words, whether the data is sorted by IP address (~2000 rows), conversations (~20,000 rows), or packets (~500,000 rows), ultimately the value taken by the response variable is directly related to the labeled role of the IP address, the IP address associated with the machine initiating a connection, or the source IP address of the packet.

Sorting the data by packets allows the most detailed analysis because the data is broken down to its most basic level. Unfortunately, because our data set does not label individual packets as being specifically malicious, we cannot achieve any more resolution beyond a particular packet's association with a maliciously labeled IP address as its originator. We could label each packet according to its source, but we would be assuming that every packet sent by a malicious IP address is a malicious packet. While this method of labeling may be sufficient for the purpose of characterizing the behavior of a malicious IP address, it is counterproductive to the purpose of classifying benign and malicious packets because it classifies benign packets from malicious IP addresses as malicious, which makes benign packets from benign IP addresses less distinguishable.

Sorting the data by IP address condenses the data into the fewest number of rows, but has the limitation of aggregating predictor data in such a way as to limit the sensitivity of prediction and classification models by decreasing the dispersion of the data. For example, at the packet level one can determine the number of bytes contained within the packet, but at the IP level, only the mean, median and standard deviation of all the packets sent by that IP address for the entire time period can be captured. While this level of data can still be useful, the processing required for collection and calculation of that data for each IP address is computationally expensive.

The compromise between these two approaches is to organize the data and collect aggregated statistics associated with the conversations (or connections) between IP addresses. This preserves more resolution than aggregating statistics by IP address, but also compresses the data more than packet-based analysis. While the aggregation is somewhat more contextual than IP-based organization, because of the labeling scheme, it is still necessary to assume that every conversation originating with a maliciously labeled IP address is a malicious conversation.

Finally, the choice of whether to code the response variable as either categorical or binary depends on the goal of the analysis. From a practical perspective, as long as the malicious conversations are distinguished from the benign, the binary response variable sufficiently separates the signal and the noise from the perspective of a network security analyst. On the other hand, if one desires to know which categories of activity are more or less reliably modeled, then the categorical coding should be used.

B. SELECTING PREDICTOR VARIABLES FOR ORDINAL LOGISTIC REGRESSION

Each of the three main categories of predictor variables has its own set of advantages. Use of the Intrusion Detection System (IDS) tool prediction values return zero false positives because they are designed to maximize specificity, which means their resulting low sensitivity makes their detection rate as a group relatively low. The Diagnostic Event variables are easy to access and collect, but perform very poorly as predictors due to their high sensitivity. IP Conversation statistics alone perform better

than the other two categories in terms of detection percentage, but not well enough to stand on their own. Our task is to determine how best to combine or layer these predictors to maximize detection rates while minimizing false positive rates.

1. Comparison of IDS Tools:

While all four IDS tools are distinct in terms of their rule sets, SNORT 2.3.2 and SNORT 2.4.1 respond identically to each other in the context of this data set, thus for our purposes they are interchangeable. Furthermore, neither tool makes any novel predictions as compared to SNORT 2.1.2 and Dragon, so for our purposes they are identical.

Table 4 shows the different prediction performance of these IDS tools taken independently. All three have a 0% false positive rate. Although SNORT 2.1.2 is the best performer, with an overall detection rate of only 21.5% it is by no means a suitable stand-alone detection tool. When combined with Dragon's unique positive predictions, they detect 29.3% of the intrusions. Table 4 presents the confusion matrix for the various tools, which compares the true and false positive rates of the various tools with ground truth.

Table 4. The highlighted diagonals show false positive (top right) and false negative (bottom left) for each of the tools. SNORT 2.1.2 and Dragon perform well against the main attacker (4) and background attackers (3), but misclassify background scanners (2) as benign (0). Although the tools do not make a prediction for chaff (1), they correctly distinguish the chaff from the main attacker, which is desirable. SNORT 2.4.1 performs worst and adds no value as a predictor since the main attacker is missed and it makes no additional detections from those already covered by the first two IDS tools.

IDS Tool Confusion Matrices

	SNORT 2.1.2				Dragon			SNORT 2.4.1		
	0	2	3	4	0	3	4	0	2	3
Ground Truth	Row%	Row %	Row %	Row %	Row %	Row %	Row %	Row %	Row %	Row %
0	100.00%	0.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%
1	100.00%	0.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	0.00%	0.00%
2	85.81%	14.19%	0.00%	0.00%	100.00%	0.00%	0.00%	94.19%	5.81%	0.00%
3	14.29%	0.00%	85.71%	0.00%	28.57%	71.43%	0.00%	35.71%	0.00%	64.29%
4	0.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%	100.00%	0.00%	0.00%

2. IP Conversation Statistics and Diagnostic Events

By themselves in a logistic regression, the IP Conversation Statistics group of variables accurately detects 66% of the threats against a categorical response variable and 68% against a binary response variable, but this higher detection comes at the expense of .61% false positive rate. Because only 1.2% of the data meets the threshold for a malicious prediction, this level of performance means that more than half of the hits resulting from the regression are false positives. It is important in this context to remember the size of the dataset and realize that .61% translates in real terms to approximately 270 false positive events per sensor, per hour.

Diagnostic events, on the other hand, provide virtually no prediction value at all. Because the alerts are spread so evenly across benign and malicious traffic alike, there is no resolution among them and therefore no added value from the variables as a group in a logistic regression.

C. ORDINAL LOGISTIC REGRESSION RESULTS: GROUPED VARIABLES

Tables 5 and 6 display modeling results for variables by category based on their performance in first-order ordinal logistic regression models. The statistics in these tables indicate that misclassification rates alone are not a good indication of the suitability of a model. First of all, because less than 1% of the data is malicious, there is little resolution among the misclassification rates. Second, the penalties for false positive predictions are the same as false negative predictions even though false negatives are less desirable. For example, Diagnostic Events—which classify every record as benign—achieve a lower misclassification rate than all but two of the models due to their higher false positive rates. Thus, we use the ratio of detection rates to false positive rates to compare predictive performance.

Table 5. The green to red trends indicate relative goodness of the variable group statistics in the context of ordinal logistic regression. The best performing individual group is IP Stats and the best combination of groups is the IP Stats and IDS predictions. The negative contribution of the Diagnostic Event variables individually and as an addition to the others in a logistic regression model is made clear by the drop in performance in the far right column.

Logistic Regression Statistics Using Ordinal Response Variable

	Diagnostic Events	IDS Pred.	IP Conv. Stats	IDS Pred., Diagnostic Events	IP Conv. Stats, Diagnostic Events	IP Conv. Stats, IDS Pred.	IP Conv. Stats, IDS Pred., Diagnostic Events
Chi2	4.52	138.57	812.58	142.48	812.58	882.81	882.81
-LL Difference	2.26	69.28	406.29	71.24	406.29	441.41	441.41
RMSE	0.0993	0.0860	0.0917	0.0860	0.2638	0.0891	0.0897
AIC	952.06	822.02	160.04	824.12	174.09	99.85	113.91
N(validation)	10890	10890	10890	10890	10890	10890	10890
Misclassification Rate							
Training	0.0062	0.0054	0.0016	0.0054	0.0017	0.0004	0.0004
Validation	0.0099	0.0074	0.0102	0.0074	0.0716	0.0092	0.0093
Validation Performance							
True Positives	0	27	65	27	65	75	74
True Negatives	10890	10863	10759	10863	10090	10749	10750
False Positives	0	0	66	0	735	66	66
False Negatives	98	71	33	71	33	23	24
Intrusion Detection%	0.0000	0.2755	0.6633	0.2755	0.6633	0.7653	0.7551
False Positive %	0.0000	0.0000	0.0061	0.0000	0.0681	0.0061	0.0061
Overall Accuracy %	1.0000	1.0000	0.9939	1.0000	0.9325	0.9939	0.9939
Detection%/False Pos%	**0.00**	**0.00**	**108.45**	**0.00**	**9.74**	**125.14**	**123.47**

Table 6. Similar to Table 5, the results from the same modeling techniques applied to a binary response variable show marginal benefits in false positive rate. The basic trends remain consistent, however. IP Stats and IDS predictions are the best group combination.

Logistic Regression Statistics Using Binary Response Variable

	Diagnostic Events	IDS Pred.	IP Conv. Stats	IDS Pred., Diagnostic Events	IP Conv. Stats, Diagnostic Events	IP Conv. Stats, IDS Pred.	IP Conv. Stats, IDS Pred., Diagnostic Events
Chi2	4.52	102.98	835.26	106.86	798.45	847.19	847.19
-LL Difference	4.52	51.47	417.63	53.43	399.22	423.60	423.60
RMSE	0.0944	0.0805	0.0890	0.0804	0.3686	0.0827	0.0891
AIC	910.43	801.99	95.72	812.09	146.58	87.80	101.86
N(validation)	10890	10890	10890	10890	10890	10890	10890
Misclass Rate							
Training	0.0062	0.0054	0.0013	0.0054	0.0015	0.0004	0.0004
Validation	0.0090	0.0065	0.0092	0.0065	0.1378	0.0081	0.0094
Validation Performance							
True Positives	0	27	67	27	64	75	75
True Negatives	10890	10863	10754	10863	9359	10750	10736
False Positives	0	0	69	0	1467	65	79
False Negatives	98	71	31	71	34	23	23
Detection%	0.0000	0.2755	0.6837	0.2755	0.6531	0.7653	0.7653
False Pos%	0.0000	0.0000	0.0064	0.0000	0.1359	0.0060	0.0073
Overall Accuracy %	1.0000	1.0000	0.9937	1.0000	0.8653	0.9940	0.9927
Detection%/False Pos%	**0.00**	**0.00**	**106.93**	**0.00**	**4.80**	**127.06**	**104.55**

Although Diagnostic Events as a group showed poor results, we considered that individual variables within the group may have some significance in the presence of variables from the other groups. To investigate this possibility as well as whether a better model could be achieved with a mix of variables from each of the three variable groups, we began with a full first order logistic regression model and conducted backwards elimination. We eliminated variables one at a time based on their individual performance in the presence of the other variables.

The variable elimination criteria were as follows: variables with $\beta = 0$ were eliminated, then those with Standard Error = 0, followed by those with the least significant chi-squared (X^2) statistics. After eliminating each variable, the resulting model performance was checked using a validation set. The process was stopped when the AICc reached a local minimum.

IV. RESULTS

A. ORDINAL LOGISTIC REGRESSION RESULTS: MIXED MODEL

Ultimately, the Diagnostic Event variables were eliminated early and were not included in the final model, however a combination of IP Conversation Statistics and IDS Tool Predictions did yield an improved model. It should be noted here that the process was repeated with the Source Port and Destination Port variables coded as ordinal as well as categorical. Although categorical coding of the port numbers is more appropriate, JMP's ordinal logistic regression algorithm would only converge with them coded as ordinal. Ultimately, both port variables were eliminated, but an argument could be made on behalf of the advantages of each coding method.

Table 7 shows the progression of the elimination steps. We use minimum AICc as a stopping rule to identify a suitable model that avoids over-fitting. In the several runs we attempted, using different variable coding methods, we noted that although the model resulting with the step after minimum AICc occurs was typically an improvement in prediction performance, it was usually also accompanied by a sharp increase in AICc, thus suggesting the improvement was due to over-fitting. The parameters we eliminated in the final model are listed below in the order of their elimination:

Step 1: NumberSecurity (Diagnostic Event)
Step 2: NumberFault (Diagnostic Event)
Step 3: Duration (IP Conversation Statistics)
Step 4: NumberInfomation (Diagnostic Event)
Step 5: NumberNotices (Diagnostic Event)
Step 6: NumberPerformance (Diagnostic Event)
Step 7: Packets Received (IP Conversation Statistics)
Step 8: Bytes Received (IP Conversation Statistics)
Step 9: NumberAlarms (Diagnostic Event)
Step 10: Number of Diagnostic Events (Diagnostic Event)
Step 11: Dragon (IDS Prediction)
Step 12: Protocol (IP Conversation Statistic)
*Step 13: Bytes (IP Conversation Statistic)
*Step 14: Packets (IP Conversation Statistic)

* Steps 13 and 14 taken to observe further performance trends only.

Table 7. At the 11th and 12th steps, the model reaches its minimum AICc (the stopping rule). After the 12th step, performance against the training set declines. Each of the four previous models (9-12) performs statistically similarly on the training set, so using AICc as the stopping criterion minimizes the likelihood of an over-fit model. The performance statistics at the bottom show that the same four models return identical results up to the 12th step. Beyond the 14th step, the model fails to converge.

Stepwise Model Statistics and Training Set Performance

	VARIABLE ELIMINATION STEPS							
	Full	**1**	**…**	**10**	**11**	**12**	**13***	**14***
ChiSquare	835.25	835.25	835.25	835.25	835.25	**835.25**	788.27	788.27
RSquare	0.9291	0.9291	0.9291	0.9291	0.9291	**0.9291**	0.8769	0.8769
Entropy R^2	-0.073	-0.023	0.1876	0.1844	0.1771	**0.1746**	0.4547	0.45
Generalized R^2	-0.078	-0.024	0.1955	0.1922	0.1847	**0.1821**	0.4671	0.4627
Mean -Log p	0.0551	0.0551	0.0417	0.0419	0.0423	**0.0424**	0.028	0.0282
RMSE	0.0843	0.0843	0.0757	0.0757	0.0757	**0.076**	0.0723	0.0719
-Loglikelihood Difference	417.62	417.62	417.62	417.62	417.62	**417.62**	394.14	394.14
AICc	109.78	105.77	93.73	91.73	89.72	**87.72**	122.67	118.67
BIC	279.46	260.69	204.41	195.03	185.65	**176.26**	166.95	148.19
Mean Abs. Dev.	0.0089	0.0089	0.0076	0.0076	0.0076	**0.0076**	0.0077	0.0076
Training Observations	11862	11862	11862	11862	11862	**11862**	11862	11862
Total Detections	89	89	89	89	89	**89**	101	101
True Positives	74	74	74	74	74	**74**	74	74
True Negatives	11773	11773	11773	11773	11773	**11773**	11761	11761
False Positives	15	15	15	15	15	**15**	27	27
False Negatives	0	0	0	0	0	**0**	0	0
Detection %	1	1	1	1	1	**1**	1	1
Accuracy %	0.9987	0.9987	0.9987	0.9987	0.9987	**0.9987**	0.9977	0.9977

Table 8. The 13th step returns the best predictive value because it is over-fit. The 12th step has sufficient accuracy and detection rates, with better protection against over-fitting.

Stepwise Model Test Set Performance

	VARIABLE ELIMINATION STEPS							
	Full	**1**	**…**	**10**	**11**	**12**	**13**	**14**
Validation Observations	10890	10890	10890	10890	10890	**10890**	10890	10890
Total Detections	163	163	150	150	150	**149**	156	151
True Positives	81	81	82	82	82	**81**	83	81
True Negatives	10710	10710	10724	10724	10724	**10724**	10727	10722
False Positives	82	82	68	68	68	**68**	73	70
False Negatives	17	17	16	16	16	**17**	15	17
Detection %	82.65	82.65	83.67	83.67	83.67	**82.65**	84.69	82.65
Accuracy %	99.09	99.09	99.23	99.23	99.23	**99.22**	99.27	99.2

Table 9. Final ordinal regression model parameters and their estimates. High multicollinearity among the variables means that many variable combinations can combine to make comparably good models, but results in high standard errors and low significance among the predictor variables.

Mixed Ordinal Regression Model Parameter Estimates

Term	Estimate (β)	Std Error	ChiSquare (X^2)	Prob>ChiSq
Intercept [0]	-18.772168	2568.1529	0	.9942
Bytes	-75.588597	15281.839	0	.9961
Bytes Sent	28.2899936	7037.365	0	.9968
Packets	103.856275	21718.97	0	.9962
Packets Sent	-83.156536	19618.83	0	.9966
SNORT 2.1.2	-934.5543	182404.1	0	.9959

Ultimately, high multicollinearity among the variables means that many variable combinations can combine to make comparably good models. For example, an alternative method of eliminating variables in order of their standard errors (from high to low) results in models of similar fit and predictive quality, but standard errors of zero and Chi-Square statistics of 10,000 (which is the default value the JMP software enters rather than "undefined" due to the zero value in the denominator of the Chi-Square calculation). Because we suspect these regression results are due to highly correlated predictor

variables, and that this is due to the lack of sophistication of the predictor variables, we apply the same method to the 1999 KDD Cup dataset in the next section in order to test whether more sophisticated predictor variables result in a logistic regression model that is both sufficiently predictive and significant.

To evaluate our predictive performance among the different categories of activity, we fit a new logistic regression using the same five predictor variables, but instead of using a binary response variable (0=Benign, 1=Malicious), we partitioned the response variable into its five ordinal categories to show clearly how well the model predicts each one. The ability to correctly predict normal (Benign) conversations is quite good, with 99.33% of normal conversations accurately predicted. Background Scanner conversations, which were confused with either Benign or Attacker conversations, were the most misclassified, at 72.94% accuracy. As the next section will describe, these are strong results when compared against the KDD-99 competition results.

Table 10. Confusion matrix results of ordinal logistic regression versus the Skaion test dataset. The bottom marginal values show the percentage of correct predictions made for that column (i.e., 99.77% of the "normal" predictions were actually "normal"). The marginal percentages on the right show the percentage of each category that were correctly identified (i.e., 99.33% of "normal" connections were predicted correctly). It is also interesting to note that the model correctly filters chaff from the categories of records that require further inspection. This occurs despite 1) the Chaff connections were designed to mimic the Attacker's signature in order to mask the Attacker's origin, and 2) there is no chaff in the training dataset so that the model can define it.

Skaion Ordinal Logistic Regression Model (Test Set)

	PREDICTED					
ACTUAL	**Benign (0)**	**Chaff (1)**	**Bkgrd Scan (2)**	**Bkgrd Atk (3)**	**Attacker (4)**	
Benign (0)	10710	0	72	0	0	**99.33%**
Chaff (1)	10	0	0	0	0	**0.00%**
Bkgrd Scan (2)	13	0	62	0	10	**72.94%**
Bkgrd Atk (3)	2	0	0	9	0	**81.82%**
Attacker (4)	0	0	0	0	2	**100.00%**
	99.77%	**0.00%**	**46.27%**	**100.00%**	**16.67%**	

Table 11. The winning results score (or "cost") 0.2331 per observation. The bottom marginal values show the percentage of correct predictions made for that column (i.e., 74.61% of the "normal" predictions were actually "normal"). The marginal percentages on the right show the percentage of each category that were correctly identified (i.e., 99.45% of "normal" connections were predicted correctly). From [19]

1999 KDD Cup Winning Results (Based on Training Dataset)

	PREDICTED					
ACTUAL	**0**	**1**	**2**	**3**	**4**	
0	60262	243	78	4	6	**99.45%**
1	511	3471	184	0	0	**83.32%**
2	5299	1328	223226	0	0	**97.12%**
3	168	20	0	30	10	**13.16%**
4	14527	294	0	8	1360	**8.40%**
	74.61%	**64.81%**	**99.88%**	**71.43%**	**98.84%**	

It is worth noting that the results posted on the KDD Cup results web page report performance against the training set only. To compare the performance of nominal logistic regression against that of the 1999 KDD Cup winner in the next section, we report performance against both the training set (Table 14) and the test set (Table 15).

B. KDD-99: VALIDATION OF LOGISTIC REGRESSION AS A CLASSIFIER

1. KDD-Cup Scoring Method

The competition participants were scored relative to one another based on the application of the cost entries in Table 12 to each misclassification made by the fitted models and divided the sum by the total number of observations (311,029) to achieve an average cost per observation. The scores range from the winning score of 0.233 to 0.9414. The distribution of the results among the 24 participants is shown in Table 13.

The winner of the 1999 KDD Cup, Bernhard Pfahringer of the Austrian Research Institute for Artificial Intelligence, achieved the results shown in Table 11 using decision trees with bagging and boosting [21]. The different modeling techniques used by the participants resulted in narrow margins of performance among the top 17 performers, but

it is worth noting within the results that 8th best submission used a simple 1-nearest neighbor classifier to achieve a score of 0.2523 [19].

Table 12. The 1999 KDD Cup competition categorized attacks based on type, similar to our method. The values in this matrix are multiplied by each number in the classification matrices used to report predictions. The objective is to minimize the cost incurred by making correct predictions, thus maximizing the sum along the diagonal so that no cost is incurred. The cost matrix establishes penalties such that the overall cost (or score) for classifying every observation as "probe" is approximately 1.0, provided categories U2R (3) and R2L (4) are relatively rare. From [19]

1999 KDD Cup Cost Matrix

	PREDICTED				
ACTUAL	**normal (0)**	**probe (1)**	**DOS (2)**	**U2R (3)**	**R2L (4)**
normal (0)	0	1	2	2	2
probe (1)	1	0	2	2	2
DOS (2)	2	1	0	2	2
U2R (3)	3	2	2	0	2
R2L (4)	4	2	2	2	0

Table 13. KDD Cup Participant Scores. The results varied widely, with a mean=0.3114, median=0.2548 and standard deviation=0.1468. The best 17 submissions all performed well, and final 7 submissions (scores of 0.2952 and greater) are considered inferior. From [19]

KDD Cup Participant Scores

0.2331 (Winner)	0.2474	0.2552	0.3344
0.2356	0.2479	0.2575	0.3767
0.2367	0.2523	0.2588	0.3854
0.2411	0.253	0.2644	0.3899
0.2414	0.2531	0.2684	0.5053
0.2443	0.2545	0.2952	0.9414

2. Logistic Regression Performance on the KDD-99 Dataset

While the previous comparison of our logistic regression results with the Skaion dataset against the results of the 1999 KDD Cup showed better performance, we determine the comparison to be biased for two main reasons: 1) our response variable categories are much broader, and 2) our dataset is far less dense with attacks. Additionally, the Chi-Squared statistics of the variables within the model were not significant.

To show the strength of logistic regression as a classifier, we used a similar backwards variable elimination process to determine a good logistic regression fit of the KDD-99 data. Because the KDD-99 dataset had such a large number of variables, we first divided the variables into four different models. Three of the sets were divided per the three feature categories described in the online task descriptions (see Appendix B), and the remaining variables comprised the fourth model. Significant variables from each model were chosen to be included in the "full" model of 34 variables. This model was then used to run backwards stepwise regression, using AICc as a stopping rule.

The regression model parameters from our resulting best model, provided in Table 14, show that the significance of the KDD-99 predictor variables is much higher than those we chose for the Skaion dataset. This both highlights the importance of variable development as well as validates the utility of logistic regression as a predictor.

Table 14. Nominal logistic regression model parameters for the KDD-99 dataset show that the significance of the variables is improved dramatically over those of the Skaion dataset.

	Log Odds for 0/4				Log Odds for 1/4				Log Odds for 2/4				Log Odds for 3/4			
Term	**Estimate**	**Std Error**	**ChiSquare**	**Prob>ChiSq**	**Estimate**	**Std Error**	**ChiSquare**	**Prob>ChiSq**	**Estimate**	**Std Error**	**ChiSquare**	**Prob>ChiSq**	**Estimate**	**Std Error**	**ChiSquare**	**Prob>ChiSq**
Intercept	16.618	1.3675	147.66	<.0001	-4.3042	1.418	9.21	0.0024	-4.9307	1.3866	12.64	0.0004	7.7133	3.6772	4.4	0.0359
count	0.0025	0.0111	0.05	0.8207	0.0091	0.0111	0.68	0.4098	0.0108	0.0111	0.94	0.3316	-0.0514	0.085	0.37	0.5454
srv_count	0.3648	0.0138	694.88	<.0001	0.3514	0.0139	639.36	<.0001	0.3568	0.0138	664.67	<.0001	0.3046	0.0753	16.37	<.0001
serror_rate	-3.7385	11.226	0.11	0.7391	-0.8053	11.231	0.01	0.9428	7.2828	11.24	0.42	0.517	-225.24	137743	0	0.9987
srv_serror_rate	4.8322	11.237	0.18	0.6672	5.5038	11.242	0.24	0.6244	-2.4976	11.251	0.05	0.8243	-237.95	184245	0	0.999
rerror_rate	-1.2876	0.6274	4.21	0.0401	4.6953	0.7056	44.28	<.0001	9.9407	0.7789	162.88	<.0001	-63.878	69489	0	0.9993
srv_rerror_rate	-0.0043	0.6861	0	0.995	-1.8642	0.8768	4.52	0.0335	-8.518	0.8362	103.78	<.0001	-156.98	100343	0	0.9988
same_srv_rate	-10.008	1.3228	57.24	<.0001	0.3431	1.3329	0.07	0.7969	-6.9594	1.3278	27.47	<.0001	-7.4418	3.5321	4.44	0.0351
diff_srv_rate	-4.1345	0.693	35.6	<.0001	7.6566	0.7308	109.77	<.0001	-5.0916	0.7438	46.86	<.0001	-6.3973	3.4364	3.47	0.0627
srv_diff_host_rate	6.5204	0.4053	258.77	<.0001	9.2252	0.4645	394.49	<.0001	-0.3839	0.4516	0.72	0.3953	2.9911	0.9283	10.38	0.0013
dst_host_count	-0.0285	0.001	858.91	<.0001	-0.0179	0.0014	153.51	<.0001	0.0168	0.0012	209.42	<.0001	-0.0154	0.0025	39.08	<.0001
dst_host_srv_count	0.027	0.0011	583.1	<.0001	0.037	0.002	359.75	<.0001	-0.002	0.0013	2.32	0.1277	-0.0118	0.0047	6.34	0.0118
dst_host_same_srv_rate	-6.8489	0.2835	583.67	<.0001	-13.768	0.5573	610.43	<.0001	3.2142	0.3731	74.22	<.0001	-6.1091	0.7469	66.91	<.0001
dst_host_diff_srv_rate	17.113	1.0968	243.42	<.0001	17.61	1.1379	239.5	<.0001	17.239	1.1168	238.25	<.0001	18.875	1.3552	194	<.0001
dst_host_same_src_port_rate	2.4641	0.282	76.34	<.0001	12.923	0.3979	1054.7	<.0001	11.905	0.2984	1592.1	<.0001	5.144	0.6859	56.25	<.0001
dst_host_srv_diff_host_rate	-0.7625	1.1369	0.45	0.5024	14.152	1.2364	131.01	<.0001	6.5764	1.1932	30.38	<.0001	2.7699	1.4038	3.89	0.0485
dst_host_serror_rate	7.5285	3.1889	5.57	0.0182	13.729	3.2349	18.01	<.0001	13.187	3.1904	17.08	<.0001	9.8995	3.2412	9.33	0.0023
dst_host_srv_serror_rate	-0.611	2.8998	0.04	0.8331	10.766	2.956	13.26	0.0003	11.595	2.8926	16.07	<.0001	6.5888	3.0899	4.55	0.033
dst_host_rerror_rate	-4.203	0.4852	75.04	<.0001	5.195	0.5351	94.27	<.0001	8.5551	0.5149	276.07	<.0001	-2.5496	1.1112	5.26	0.0218
dst_host_srv_rerror_rate	-3.7988	0.445	72.89	<.0001	-0.3508	0.7167	0.24	0.6245	-2.7576	0.5346	26.61	<.0001	1.0217	0.9286	1.21	0.2712
logged_in	0.256	0.0275	86.95	<.0001	-2.8125	0.3687	58.19	<.0001	3.6662	0.0712	2654.8	<.0001	2.4158	0.3253	55.15	<.0001
is_guest_login	-2.9538	0.1857	253.04	<.0001	-23.214	187745	0	0.9999	4.5302	0.1936	547.36	<.0001	-2.7602	0.6608	17.45	<.0001
num_file_creations	1.9007	0.8664	4.81	0.0282	-22.457	234978	0	0.9999	-4.0126	1.2211	10.8	0.001	1.9221	0.8665	4.92	0.0265
is_host_login	-7.2283	1.4508	24.82	<.0001	-28.581	1E+06	0	1	-6.7878	1.5003	20.47	<.0001	-2.3971	1.2229	3.84	0.05

As with the Skaion dataset, we attempted both nominal and ordinal logistic regression, and in the case of KDD-99 we found nominal coding of the response variables led to earlier convergence of the model (i.e., model converged after fewer variable elimination steps). Our results against the training set, shown in Table 15, represent a 7.8% improvement over the winning results. Our performance against the test dataset, shown in Table 16, represents a 66.8% improvement over the winner's training set results. Again, because the test set results were not posted, we can only compare against the winner's training set.

Table 15. Cost score using the 1999 KDD Cup cost matrix on results of ordinal logistic regression versus the KDD Cup training dataset is 0.2150 (7.8% lower than the winning score of 0.2331).

Nominal Logistic Regression Results vs. 1999 KDD Cup Training Dataset

	PREDICTED					
ACTUAL	**0**	**1**	**2**	**3**	**4**	
0	50997	189	1338	4	8065	**84.16%**
1	159	3890	104	0	13	**93.37%**
2	5490	100	225814	0	51	**97.56%**
3	2	0	15	17	4	**44.74%**
4	9043	0	36	4	5644	**38.32%**
	77.63%	**93.08%**	**99.34%**	**68.00%**	**40.97%**	

Table 16. Cost score using the 1999 KDD Cup cost matrix on results of ordinal logistic regression versus the KDD Cup test dataset is 0.0774. Again, we cannot compare these results to KDD Cup test set results, since they were not posted.

Nominal Logistic Regression Results vs. 1999 KDD Cup Test Dataset

	PREDICTED					
ACTUAL	**0**	**1**	**2**	**3**	**4**	
0	78249	9118	3085	47	6779	**80.44%**
1	124	3674	301	0	8	**89.46%**
2	2679	333	388278	0	188	**99.18%**
3	32	0	22	3	4	**4.92%**
4	62	1	1033	1	0	**0.00%**
	96.43%	**27.99%**	**98.87%**	**5.88%**	**0.00%**	

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

1. Individual Tools and Methods are Inadequate by Themselves

There are many approaches to both signature and anomaly detection in network security. Each category and each tool have their strengths and weaknesses and all must be responsive to the dynamic nature of the threat. Since requiring analysts to continue stacking detection tools or rule sets indefinitely is untenable, methods of combining responses generated by these tools should be used to help streamline analysis. Our method showed the merits of one approach which drastically improved the predictive performance of an intrusion detection tool and basic IP conversation statistics.

Furthermore, no tool or method we studied was 100% effective against all threats. Although ranking methods can help prioritize investigation of more immediate or more dangerous *known* threats, it will do nothing to protect against *unknown* threats (novel or zero-day attacks). Thus, learning systems that adjust their definition of "normal" and take into account network environment statistics should be improved. Good examples of these types of systems fall under the anomaly detection category.

2. With Good Predictors, Simple Methods Can Be Very Powerful

Our classification of the Skaion dataset's threats was accomplished with an elementary set of significantly collinear predictor variables, yet they performed comparatively well in a logistic regression. The validation of that performance against a much more complex dataset was due to the high quality of the predictor variables available in that dataset, which were more contextual and more independent than those we chose for Skaion. Of course, balance must be sought between the goals of constructing powerful predictor variables and minimizing the computational overhead associated with their construction.

3. Experimental Network Design Needs to Consider Analysis

One of the most important findings on which we can report from this study is the difficulty associated with data exploration in this field. First, the amount of data available in even a simple scenario conducted over a short period of time can be difficult to manage. Second, understanding the context of the data can be problematic for those unfamiliar with network design and security. Third, even with that background knowledge, the ground truth must be both detailed and clear in order to avoid wasting analysis opportunities and making false assumptions.

The Skaion dataset was created using novel methods of network traffic generation and emulation while maintaining data anonymity, which is crucial to its academic utility. Unfortunately, due to imprecise labeling, disorganized file structure, and lack of preprocessing, that utility is inaccessible to analysts without a strong computer science background. Conversely, the 1999 KDD Cup dataset was presented to participants with enough pre-processing complete that statisticians and data miners who were not computer scientists were able to participate. Although it has its own shortcomings [16], this accessibility is why that dataset is still used as an industry standard for academic study.

We recommend that network emulators create a set of companion files based on their scenarios that provide analysis opportunities similar to the 1999 KDD Cup dataset. The types of predictor variables could be identical, or they could add more that they have found to be good performers. While the raw data is valuable, there are many dimensions to the network security problem that do not require getting down to a deep packet inspection level.

Additionally, Skaion should add more intrusion detection predictions similar to the four they include, but that vary more in their responses. The intent should be to simulate the data overload an analyst might experience from having too many inputs to analyze in a reasonable amount of time in order to allow study on how to compress those incoming signals to minimize duplication of effort on common events while maintaining enough resolution to detect rare events. These classic signal-to-noise ratio problems are fertile ground for human system interface, data visualization, and data mining analysts of

all flavors, many of whom may not have the computer science background necessary to "break the code" of the Skaion data set's file system.

B. RECOMMENDATIONS FOR FUTURE WORK

1. Further Exploitation of the Skaion Dataset

Since we feel we have only scratched the surface of what the Skaion dataset has to offer, we recommend that further effort be made to extract better predictor variables in order to test more robust methods. Our desire to work primarily in a Microsoft Windows environment limited full assimilation of the ground truth and network flow data included in the dataset. Being able to manipulate this data using the Unix programs for which they were intended would have allowed us two advantages:

1. Variables created from the IDS tool alert logs could be matched to each individual conversation, as opposed to just their predictions of IP address role, as we used them;
2. Network flow statistics could be accessed and new variables created.

This work would require either an analyst with a network security background, or collaboration with the Skaion Corporation to accomplish the necessary pre-processing of the dataset.

2. Use the 1999 KDD Cup Dataset to Test Better Data Fusion Methods

The accessibility of the 1999 KDD Cup dataset makes it a perfect candidate as a foundation for testing new modeling techniques. Since the competition occurred over 13 years ago, the remaining opportunities for first-order detection models are probably few. However, since data visualization and fusion are perhaps an even bigger challenge now due to the increased complexity and sophistication of the threat, opportunities here may actually be increasing. Using the KDD Cup dataset as a backdrop, and multiple intrusion detection systems of varied detection abilities, more work could be done to improve prioritization of threat response as well as optimize both computational and analytical workload involved with intrusion detection.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDICES

A. MACROS USED TO SORT AND CLEAN EXPORTED PACKET DATA

The following Visual Basic for Applications (VBA) code was implemented with Microsoft Excel to pre-sort and clean the raw data. The runtime to process a dataset of 200 kilobytes is approximately five minutes.

```
Sub AddErrorlClassColumns()
'
' AddErrorlClassColumns Macro
' Adds error sum columns and classification columns
'

'
    Sheets("Error Key").Select
    Range("A2:A22").Select
    Selection.Copy
    Sheets("IP Stats").Select
    Range("AH1").Select
    Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:= _
        False, Transpose:=True
    Columns("A:A").EntireColumn.AutoFit
    Columns("B:B").EntireColumn.AutoFit
    ActiveWindow.SmallScroll Down:=96
    Columns("B:B").Select
    Application.CutCopyMode = False
    Selection.ClearContents
    Range("B1").Select
    ActiveCell.FormulaR1C1 = "IP Classification"
    Range("B2").Select
End Sub

Sub DeleteCountryRows()
'
' DeleteCountryRows Macro
'

'
    Selection.AutoFilter
    ActiveCell.FormulaR1C1 = "=IF(RC[12]=0,0,1)"
    Range("B2").Select
    Selection.AutoFill Destination:=Range("B2:B2270"), Type:=xlFillDefault
    Range("B2:B2270").Select

    Range("BC1").Select
    ActiveCell.FormulaR1C1 = "Country"
    Range("BC3").Select
    ActiveCell.FormulaR1C1 = "N/A"
    Range("BC4").Select
    ActiveCell.FormulaR1C1 = "=IF(RC[-53]=1,R[-1]C,IF(RC[-53]=0,RC[-54]))"
```

```
    Range(“BC4”).Select
    Selection.AutoFill Destination:=Range(“BC4:BC972”), Type:=xlFillDefault
    Range(“BC4:BC972”).Select
    Selection.AutoFill Destination:=Range(“BC4:BC2270”), Type:=xlFillDefault
    Range(“BC4:BC2270”).Select

    Range(“E25”).Select
    Columns(“BC:BC”).Select
    Selection.Copy
    Columns(“BD:BD”).Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Columns(“BC:BC”).Select
    Application.CutCopyMode = False

    Rows(“1:1”).Select
    ActiveWorkbook.Worksheets(“IP Stats”).AutoFilter.Sort.SortFields. _
        Clear
    ActiveWorkbook.Worksheets(“IP Stats”).AutoFilter.Sort.SortFields. _
        Add Key:=Range(“B1”), SortOn:=xlSortOnValues, Order:=xlDescending, _
        DataOption:=xlSortTextAsNumbers
    With ActiveWorkbook.Worksheets(“IP Stats”).AutoFilter.Sort
        .Header = xlYes
        .MatchCase = False
        .Orientation = xlTopToBottom
        .SortMethod = xlPinYin
        .Apply
    End With

End Sub

Sub FillErrorClassCols()
'
' FillErrorClassCols Macro
' Fill all the error columns and rows as well as classification column
'

'
    Range(“B2”).Select
    ActiveCell.FormulaR1C1 = “=LOOKUP(RC[-1],'ip-key'!C[-1],'ip-key'!C)”
    Range(“B2”).Select
    Selection.AutoFill Destination:=Range(“B2:B2390”), Type:=xlFillDefault
    Range(“AH2”).Select
    ActiveCell.FormulaR1C1 = _
        “=COUNTIFS('Diagnosis Events'!C7,'IP Stats'!RC1,'Diagnosis Events'!C1,'IP
Stats'!R1C)”
    Range(“AH2”).Select
    Selection.AutoFill Destination:=Range(“AH2:BB2”), Type:=xlFillDefault
    Range(“AH2:BB2”).Select
    Range(“AH2:BB2”).Select
    Range(“BB2”).Activate
    Selection.AutoFill Destination:=Range(“AH2:BB2439”), Type:=xlFillDefault
    Range(“AH2:BB2439”).Select
    Range(“AW2439”).Select
```

```
    Application.Calculation = xlManual
    Range("A1").Select

    Sheets.Add After:=Sheets(Sheets.Count)
    Sheets("IP Stats").Select
    Range("A1:BC2439").Select
    Range("A1753").Activate
    Selection.Copy
    Sheets("Sheet4").Select
    Range("A1").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Sheets("IP Stats").Select
    Application.CutCopyMode = False
    ActiveWindow.SelectedSheets.Delete
    Application.Calculation = xlAutomatic

End Sub

Sub DeleteExtraDataTags()
'
' DeleteExtraDataTags Macro
' Clean up errant KBs
'

'
    Columns("C:C").Select
    Selection.TextToColumns Destination:=Range("C1"), DataType:=xlDelimited, _
        TextQualifier:=xlDoubleQuote, ConsecutiveDelimiter:=True, Tab:=False, _
        Semicolon:=False, Comma:=False, Space:=True, Other:=False, FieldInfo _
        :=Array(Array(1, 1), Array(2, 9)), TrailingMinusNumbers:=True
    Columns("E:E").Select
    Selection.TextToColumns Destination:=Range("E1"), DataType:=xlDelimited, _
        TextQualifier:=xlDoubleQuote, ConsecutiveDelimiter:=True, Tab:=False, _
        Semicolon:=False, Comma:=False, Space:=True, Other:=False, FieldInfo _
        :=Array(Array(1,      1),      Array(2,      9),      Array(3,      9)),
TrailingMinusNumbers:=True
    Columns("F:F").Select
    Selection.TextToColumns Destination:=Range("F1"), DataType:=xlDelimited, _
        TextQualifier:=xlDoubleQuote, ConsecutiveDelimiter:=True, Tab:=False, _
        Semicolon:=False, Comma:=False, Space:=True, Other:=False, FieldInfo _
        :=Array(Array(1,      1),      Array(2,      9),      Array(3,      9)),
TrailingMinusNumbers:=True
    Columns("H:H").Select
    Selection.TextToColumns Destination:=Range("H1"), DataType:=xlDelimited, _
        TextQualifier:=xlDoubleQuote, ConsecutiveDelimiter:=True, Tab:=False, _
        Semicolon:=False, Comma:=False, Space:=True, Other:=False, FieldInfo _
        :=Array(Array(1, 1), Array(2, 9)), TrailingMinusNumbers:=True
    Columns("J:J").Select
    Selection.TextToColumns Destination:=Range("J1"), DataType:=xlDelimited, _
        TextQualifier:=xlDoubleQuote, ConsecutiveDelimiter:=True, Tab:=False, _
        Semicolon:=False, Comma:=False, Space:=True, Other:=False, FieldInfo _
        :=Array(Array(1, 1), Array(2, 9)), TrailingMinusNumbers:=True
    Columns("C:J").Select
    Columns("C:J").EntireColumn.AutoFit
```

```
    Range(“C1”).Select
    ActiveCell.FormulaR1C1 = “Kilobytes”
    Range(“E1”).Select
    ActiveCell.FormulaR1C1 = “Kilobits Per Second”
    Range(“F1”).Select
    ActiveCell.FormulaR1C1 = “Kilobytes Per Second”
    Range(“H1”).Select
    ActiveCell.FormulaR1C1 = “Kilobytes Received”
    Range(“J1”).Select
    ActiveCell.FormulaR1C1 = “Kilobytes Sent”
    Range(“J2”).Select
End Sub
Sub AddClassCols()
'
' AddClassCols Macro
'

'
    Columns(“C:C”).Select
    Selection.Insert Shift:=xlToRight, CopyOrigin:=xlFormatFromLeftOrAbove
    Columns(“D:D”).Select
    Selection.Insert Shift:=xlToRight, CopyOrigin:=xlFormatFromLeftOrAbove
    Range(“C1”).Select
    ActiveCell.FormulaR1C1 = “Binary IP Classification”
    Range(“D1”).Select
    ActiveCell.FormulaR1C1 = “Ordinal IP Classification”
    Range(“C2”).Select
    ActiveCell.FormulaR1C1 = _
        “=IF(RC[-1]=““ATTACKER““,1,IF(RC[-1]=““BACKGROUND    ATTACKER““,1,IF(RC[-
1]=““BACKGROUND SCANNER““,1,0)))”
    Range(“C2”).Select
    Selection.Copy
    Range(“C2”).Select
    Range(Selection, Selection.End(xlDown)).Select
    Application.CutCopyMode = False
    Selection.FillDown
    Range(“D2”).Select
    ActiveCell.FormulaR1C1 = _
        “=IF(RC[-2]=““VICTIM““,0,IF(RC[-2]=““SERVER““,0,IF(RC[-
2]=““CLIENT““,0,IF(RC[-2]=““BACKGROUND          SCANNER““,1,IF(RC[-2]=““BACKGROUND
ATTACKER““,2,IF(RC[-2]=““ATTACKER““,3,0))))))”
    Range(“D2”).Select
    Selection.Copy
    Range(Selection, Selection.End(xlDown)).Select
    Application.CutCopyMode = False
    Selection.FillDown
    Range(“A1”).Select
    Calculate
End Sub
```

B. KDD CUP 1999 DATASET DETAILS (FROM THE KDD-CUP 1999 WEBPAGE)[19]

This document is adapted from the paper *Cost-based Modeling and Evaluation for Data Mining With Application to Fraud and Intrusion Detection: Results from the JAM Project* by Salvatore J. Stolfo, Wei Fan, Wenke Lee, Andreas Prodromidis, and Philip K. Chan.

1. INTRUSION DETECTOR LEARNING

Software to detect network intrusions protects a computer network from unauthorized users, including perhaps insiders. The intrusion detector learning task is to build a predictive model (i.e., a classifier) capable of distinguishing between ``bad'' connections, called intrusions or attacks, and ``good'' normal connections.

The 1998 DARPA Intrusion Detection Evaluation Program was prepared and managed by MIT Lincoln Labs. The objective was to survey and evaluate research in intrusion detection. A standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment, was provided. The 1999 KDD intrusion detection contest uses a version of this dataset.

Lincoln Labs set up an environment to acquire nine weeks of raw TCP dump data for a local-area network (LAN) simulating a typical U.S. Air Force LAN. They operated the LAN as if it were a true Air Force environment, but peppered it with multiple attacks.

The raw training data was about four gigabytes of compressed binary TCP dump data from seven weeks of network traffic. This was processed into about five million connection records. Similarly, the two weeks of test data yielded around two million connection records.

A connection is a sequence of TCP packets starting and ending at some well-defined times, between which data flows to and from a source IP address to a target IP address under some well-defined protocol. Each connection is labeled as either normal, or as an attack, with exactly one specific attack type. Each connection record consists of about 100 bytes.

Attacks fall into four main categories:

DOS: denial-of-service, e.g. syn flood;

R2L: unauthorized access from a remote machine, e.g. guessing password;

U2R: unauthorized access to local superuser (root) privileges, e.g., various ``buffer overflow'' attacks;

Probing: surveillance and other probing, e.g., port scanning.

It is important to note that the test data is not from the same probability distribution as the training data, and it includes specific attack types not in the training

data. This makes the task more realistic. Some intrusion experts believe that most novel attacks are variants of known attacks and the “signature” of known attacks can be sufficient to catch novel variants. The datasets contain a total of 24 training attack types, with an additional 14 types in the test data only.

2. DERIVED FEATURES

Stolfo et al. defined higher-level features that help in distinguishing normal connections from attacks. There are several categories of derived features.

The ``same host" features examine only the connections in the past two seconds that have the same destination host as the current connection, and calculate statistics related to protocol behavior, service, etc.

The similar ``same service" features examine only the connections in the past two seconds that have the same service as the current connection.

“Same host” and “same service” features are together called time-based traffic features of the connection records.

Some probing attacks scan the hosts (or ports) using a much larger time interval than two seconds, for example once per minute. Therefore, connection records were also sorted by destination host, and features were constructed using a window of 100 connections to the same host instead of a time window. This yields a set of so-called host-based traffic features.

Unlike most of the DOS and probing attacks, there appear to be no sequential patterns that are frequent in records of R2L and U2R attacks. This is because the DOS and probing attacks involve many connections to some host(s) in a very short period of time, but the R2L and U2R attacks are embedded in the data portions
of packets, and normally involve only a single connection.

Useful algorithms for mining the unstructured data portions of packets automatically are an open research question. Stolfo et al. used domain knowledge to add features that look for suspicious behavior in the data portions, such as the number of failed login attempts. These features are called ``content" features.

A complete listing of the set of features defined for the connection records is given in the three tables below. The data schema of the contest dataset is available in machine-readable form.

feature name	*description*	*type*
duration	length (number of seconds) of the connection	continuous
protocol_type	type of the protocol, e.g. tcp, udp, etc.	discrete
service	network service on the destination, e.g., http, telnet, etc.	discrete
src_bytes	number of data bytes from source to destination	continuous
dst_bytes	number of data bytes from destination to source	continuous
flag	normal or error status of the connection	discrete
land	1 if connection is from/to the same host/port; 0 otherwise	discrete
wrong_fragment	number of “wrong" fragments	continuous
urgent	number of urgent packets	continuous

Table 17. Basic features of individual TCP connections.

feature name	*description*	*type*
hot	number of “hot" indicators	continuous
num_failed_logins	number of failed login attempts	continuous
logged_in	1 if successfully logged in; 0 otherwise	discrete
num_compromised	number of “compromised" conditions	continuous
root_shell	1 if root shell is obtained; 0 otherwise	discrete
su_attempted	1 if “su root" command attempted; 0 otherwise	discrete
num_root	number of “root" accesses	continuous
num_file_creations	number of file creation operations	continuous
num_shells	number of shell prompts	continuous
num_access_files	number of operations on access control files	continuous
num_outbound_cmds	number of outbound commands in an ftp session	continuous
is_hot_login	1 if the login belongs to the ``hot" list; 0 otherwise	discrete
is_guest_login	1 if the login is a ``guest"login; 0 otherwise	discrete

Table 18. Content features within a connection suggested by domain knowledge.

feature name	description	type
count	number of connections to the same host as the current connection in the past two seconds	continuous
	Note: The following features refer to these same-host connections.	
serror_rate	% of connections that have ``SYN'' errors	continuous
rerror_rate	% of connections that have ``REJ'' errors	continuous
same_srv_rate	% of connections to the same service	continuous
diff_srv_rate	% of connections to different services	continuous
srv_count	number of connections to the same service as the current connection in the past two seconds	continuous
	Note: The following features refer to these same-service connections.	
srv_serror_rate	% of connections that have ``SYN'' errors	continuous
srv_rerror_rate	% of connections that have ``REJ'' errors	continuous
srv_diff_host_rate	% of connections to different hosts	continuous

Table 19. Traffic features computed using a two-second time window.

LIST OF REFERENCES

[1] (2011, March 24) *Proceedings of the special meeting on mission assurance: analysis for cyber operations* [Online]. Available: http://www.mors.org/UserFiles/file/2011%20Cyber%20Assurance/Outbriefs/Cyber%20II%20Final%20Report_March%202011.pdf

[2] (2012, June 11). *80th MORS symposium quick reference program schedule* [Online]. Available: http://www.mors.org/UserFiles/file/80th%20Symposium/MORS%20QRPS%202012.pdf

[3] (2012, July 29). *Cyber academic group curriculum catalog* [Online]. Available: http://www.mors.org/UserFiles/file/2011%20Cyber%20Assurance/Outbriefs/Cyber%20II%20Final%20Report_March%202011.pdf

[4] N. Falliere, L. O. Murchu, and E. Chien, "W32.stuxnet dossier, version 1.4," Symantec Corp., Cupertino, CA, Tech. Rep. Version 1.4, Feb. 11, 2011.

[5] D. McMorrow, "JASON technical report: Science of cyber-security," The MITRE Corp., McLean, VA, Tech. Rep. 13109022, 2010.

[6] S. W. Korns and J. E. Kastenberg, "Georgia's cyber left hook," *Parameters*, vol. 38, pp. 60–76, Jan. 2008.

[7] M. Lake, (2012, May 10). *DARPA contract N66001-04-C-8006* [Online]. Available: https://e-commerce.spawar.navy.mil/command/02/acq/navhome.nsf/homepage?readform&db=navbusopor.nsf&whichdoc=D6D08D5E39F6FBE988256DE50061C3A6&editflag=0

[8] (2003, October 01). *Press release: Skaion awarded ARDA grant for information security research* [Online]. Available: http://www.skaion.com/news/rel20031001.html

[9] "Cyberspace policy review," The White House, Washington, D.C., Policy Review, 2009.

[10] M. G. Mullen, "Joint publication 6-0: Joint communications system," U.S. Department of Defense, Washington, D.C., Joint Publication JP 6-0, 2010.

[11] P. Szor, *The Art of Computer Virus Research and Defense.* Hagerstown, MD: Addison-Wesley, 2005.

[12] Y. Wang, *Statistical Techniques for Network Security: Modern Statistically-Based Intrusion Detection and Protection.* New Haven, CT: IGI Global, 2008.

[13] P. Gogoi, D. Bhattacharyya, B. Borah, and J. K. Kalita, "A survey of outlier detection methods in network anomaly identification," *The Computer J.*, vol. 54, pp. 570, Apr. 2011.

[14] M. Mahoney and P. Chan. "An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection." in *Proceedings of the 6th Int. Symp on Recent Advances in Intrusion Detection*, Pittsburgh, PA, 2003.

[15] K. Yamanishi, J. Takeuchi, G. Williams, and P. Milne, "On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms," *Data Mining and Knowledge Discovery*, vol. 8, pp. 275-300, May 2004.

[16] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD cup 99 data set," in *Proc. of the 2009 IEEE Symp. on Computational Intelligence in Security and Defense Applications (CISDA 2009)*, Ottowa, ON, 2009.

[17] Y. Wang and I. Kim, "A bootstrap-based simple probability model for classifying network traffic and detecting network intrusion," *Security J.*, vol. 21, pp. 278—290, Aug. 2008.

[18] S. Aftergood. (2007, February 13). *Open source information system (OSIS)* [Online]. Available: http://www.fas.org/irp/program/disseminate/osis.htm

[19] S. Hettich and S. D. Bay. (2005, September 9). *The UCI KDD archive* [Online]. Available: http://kdd.ics.uci.edu

[20] (2012, September 10). *JMP support site* [Online]. Available: http://www.jmp.com/support/

[21] B. Pfahringer, "Winning the KDD99 classification cup: Bagged boosting," *ACM SIGKDD Explorations Newsletter*, vol. 1, pp. 65—66, Jan. 2000.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Robert F. Dell
   Naval Postgraduate School
   Monterey, California

4. Lyn R. Whitaker
   Naval Postgraduate School
   Monterey, California

5. Harrison C. Schramm
   Naval Postgraduate School
   Monterey, California

6. Alexander Kott
   Network Science Division, Army Research Laboratory
   Adelphi, Maryland

7. Richard E. Harang
   Network Science Division, Army Research Laboratory
   Adelphi, Maryland

8. RADM (Ret) Andy Singer
   Information Dominance Center for Excellence
   Monterey, California

9. Dr. Jerry Smith
   Assessments Division, Chief of Naval Operations Staff
   Arlington, Virginia

10. Terrence Champion
    Skaion Corporation
    North Chelmsford, Massachusetts

11. William F. Major
    PATROL SQUADRON THIRTY (VP-30)
    Jacksonville, Florida